Robust face alignment and partial face recognition

Weng, Renliang

2016

Weng, R. (2016). Robust face alignment and partial face recognition. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/66026>

<https://doi.org/10.32657/10356/66026>

Robust Face Alignment and Partial Face Recognition

Renliang Weng

School of Electrical and Electronic Engineering

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

2015

To my parents

Acknowledgments

First and foremost, I would like to thank my supervisor Prof. Yap-Peng Tan for his edifying guidance on my research. Under his supervision, I have not only learned the principles in conducting research, but also been influenced by his prudent scientific attitude. He also encourages me to broaden my horizon by thinking out of the box, to link between different research areas.

I would also like to express my gratitude to Dr. Jiwen Lu for his patient and generous advice. He helped me greatly in developing my ideas, tackling with mathematical problems, organizing and writing papers, conducting experiments, *etc.*. Without his help, this work would not have come to fruition.

I would like to acknowledge the fund and Ph.D. grant from the Rapid Rich Object Search (ROSE) Lab, Nanyang Technological University, Singapore. I have had pleasant time in studying and researching in ROSE lab. ROSE has the cutting-edge hardware facility, which enables me to conduct experiments efficiently.

I would like to extend my thanks to Dr. Yuwei Wu and Junlin Hu for their valuable suggestions and discussion on various research problems in the process of my research. I am also thankful to Yan Wang, Zhen Zuo, Lifan Zhao, Yangtao Wang, Dr. Zhenwei Miao, among many other kind people who have helped me a lot.

Finally, I would like to thank my parents and Jie Yan, without their support and encouragement, this thesis would not be possible.

Contents

Acknowledgments	i
Abstract	vi
List of Figures	ix
List of Tables	xx
List of Abbreviations	xxii
List of Notations	xxiii
1 Introduction	1
1.1 Background	1
1.2 Motivation	5
1.3 Main Contributions of the Thesis	7
1.4 Organization of the Thesis	9
2 Related Works	11
2.1 Face Alignment	11
2.1.1 Face Landmark Detection	12

2.1.2	Appearance Similarity Maximization	17
2.1.3	Discussion	19
2.2	Partial Face Recognition	20
2.2.1	Robust Face Recognition	21
2.2.2	Point Set Matching and Graph Matching	28
2.2.3	Discussion	30
2.3	Conclusions	30
3	Holistic Face Alignment Using CDAN	32
3.1	Introduction	32
3.2	Cascaded Deep Auto-encoder Network	34
3.2.1	Global Exemplar-based Deep Auto-encoder Network	35
3.2.2	Localized Deep Auto-encoder Network	42
3.2.3	Difference with CFAN	48
3.3	Implementation Details	49
3.4	Experiments	50
3.4.1	Datasets and Experimental Settings	52
3.4.2	Evaluation of GEDAN	53
3.4.3	Evaluation of LDAN	55
3.4.4	Evaluation of Cascaded LDANs	58
3.4.5	Comparison with the State-of-the-Art Methods	58
3.4.6	Computational Time	62
3.5	Conclusions	63
4	Holistic Face Alignment Using HiCNN	64

4.1	Introduction	65
4.2	Hierarchical Convolutional Neural Network	68
4.2.1	Global CNN	69
4.2.2	Part-based CNN	73
4.2.3	Patch-based CNN	78
4.3	Implementation Details	81
4.4	Experiments	81
4.4.1	Datasets and Experimental Settings	82
4.4.2	Investigation of the Global CNN	84
4.4.3	Investigation of the part-based CNN	88
4.4.4	Investigation of the patch-based CNN	92
4.4.5	Comparison with state-of-the-art methods	94
4.4.6	Computational Time	101
4.5	Conclusions	102
5	Partial Face Alignment and Recognition	104
5.1	Introduction	105
5.2	Robust Point Set Matching	107
5.2.1	Feature Extraction	108
5.2.2	Keypoint Selection	109
5.2.3	Point Set Matching	109
5.2.4	Point-Set Distance	117
5.2.5	Parameter Settings	118
5.3	Experiments	121

5.3.1	Data Sets	122
5.3.2	Baseline algorithms	123
5.3.3	Evaluation of the Point-Set Distance Metric	124
5.3.4	Partial Face Alignment Performance	126
5.3.5	Partial Face Recognition Performance	128
5.3.6	RPSM vs. Holistic face recognition approach	140
5.3.7	Discussions	144
5.3.8	Computational Time	145
5.4	Conclusion	146
6	Conclusions and Future Research	148
6.1	Conclusions	148
6.2	Recommendations for Future Research	150
	Author's Publications	152
	Bibliography	153

Abstract

Face alignment and face recognition are two fundamental problems in the facial analysis community. For face alignment, it forms the basis for the accurate face recognition, age estimation, and facial expression recognition. For face recognition, it has been widely applied in various practical scenarios such as access control system, massive surveillance, human computer interaction, *etc.*. There mainly exist two lines of works in these two fields, namely *holistic face alignment and recognition*, and *partial face alignment and recognition*. Numerous holistic face alignment and recognition works have been proposed and recent state of the arts have surpassed human’s recognition capability on the challenging LFW dataset. One of the major challenges of this area lies on designing robust holistic face alignment method which can accurately detect landmarks from faces with large facial poses. On the contrary, relatively few works have been proposed to deal with partial face alignment and recognition, and they have achieved limited success. In this thesis, we aim to advance the holistic face alignment and contribute to the field of partial face alignment and recognition. In particular, for the holistic face alignment, we devise two deep learning based approaches which are capable of estimating facial landmark positions with great robustness and high accuracy. In terms of the partial face alignment and recognition, we present an approach based on robust feature set matching, which achieves partial face alignment and recognition jointly in a single framework.

For the holistic face alignment, we are interested in the facial landmark detection problem. The mainstream face landmark detection approaches consist of a pose initialization stage and a pose update step. The pose initialization step derives an initial pose for face alignment. Since the face landmark detection is a highly non-convex problem, this initial pose largely determines the local basin where the final solution arrives. The pose update stage then locally refines the initial pose to achieve high alignment

accuracy. Both of these two steps are critical for achieving robust and accurate face alignment performance. In our first work, to improve the robustness of the pose initialization step against large pose variations, we devise a Global Exemplar-based Deep Auto-encoder Network (GEDAN), whose top regression layer deploys several exemplars to assist pose estimation. For the pose update stage, we design a series of Localized Deep Auto-encoder Networks (LDAN). Specifically, its first layer consists of individual Local Auto-Encoders (LAEs). Each LAE aims to extract pose-related features from its corresponding local patch. The outputs of these LAEs are then directly fed into their corresponding local regressors. In addition, these outputs are concatenated into a global feature vector which is further encoded by several layers of auto-encoders to preserve the global facial structure. By assembling GEDAN and several LDANs together in a coarse-to-fine way, our approach achieves superior alignment accuracy with real-time speed. We term this network ensemble as Cascaded Deep Auto-encoder Networks (CDAN).

While CDAN works well on near-upright faces, it's incapable of detecting landmarks from arbitrarily rotated facial images. To this end, we leverage the strength of the Convolutional Neural Networks (CNN) and devise a Hierarchical CNN (HiCNN) cascade. In particular, HiCNN consists of a global CNN, a part-based CNN and a patch-based CNN. The global CNN generates a preliminary four-landmark configuration from the low-resolution facial image. Based on this preliminary result, landmark positions are estimated by the part-based CNN based on the corresponding facial parts on a larger resolution. Lastly, the patch-based CNN refines the landmark positions from the view of pose-indexed patches at the highest resolution. Extensive experiments on three benchmarks show that the proposed HiCNN can accurately detect landmarks from facial images with *arbitrary* in-plane rotation, large scale variations and random face shifts.

Both CDAN and HiCNN are holistic face alignment methods, they may fail if the facial image is an arbitrary facial patch. In realistic scenarios, however, faces might be severely occluded or randomly cropped, resulting in partial faces. It's desirable to automatically align these partial faces to holistic facial image and subsequently recognize them. To this end, we propose a new partial face recognition approach named Robust Point Set Matching (RPSM) by using feature set matching, which is able to align partial face patches to holistic gallery faces automatically and is robust to occlusions and illumination changes. Given each gallery image and probe face patch, we first detect

keypoints and extract their local features. Then, the RPSM matches the extracted local feature sets by minimizing the geometric and textural difference. Lastly, the similarity of two faces is converted as the distance between two feature sets. The matching problem is formulated in a linear programming framework; hence, constraint of affine transformation can be easily applied to restrain from unrealistic face warping. The proposed RPSM achieves superior results both on partial face alignment and partial face recognition on four public face datasets.

List of Figures

1.1	Partial face examples. (a) Three partial face patches (in the red ellipse) are from the LFW dataset [1] which are occluded by heads; (b) Face occluded by sunglasses. (c) Arbitrary partial face patch.	2
1.2	The system flow of a standard holistic/partial face recognition approach. In this thesis, we focus on the problems of holistic face alignment, partial face alignment and partial face recognition. These three topics are accentuated in blue color.	3
1.3	Two holistic face alignment examples. (a) Facial landmark detection result achieved by our CDAN, where the green dots represent the detected 68 facial landmarks. (b) Appearance similarity maximization based face alignment result, where facial images in the first row are the unaligned faces with corruption and the faces in the second row are the aligned faces. Facial images of (b) are referenced from the paper RASL [2]. . .	4
2.1	The Global-SAN and the Local-SAN employed in the CFAN model. . .	16

3.1	The pipeline of our Cascaded Deep Auto-encoder Networks (CDAN) for facial landmark detection. The facial landmarks positions are gradually refined as the image passes through GEDAN and several LDANs from the left to the right. Meanwhile facial image resolution is increased as well, such that more details are incorporated for face alignment. Specifically, we have $\mathbf{S}_{i+1} = \mathbf{S}_i + \Delta\mathbf{S}_{i+1}$, where $\mathbf{S}_0$ is the shape prediction of GEDAN.	33
3.2	The architecture of our Global Exemplar-based Deep Auto-encoder Network (GEDAN). The first three layers are auto-encoder layers, which progressively encode the raw facial image vector $\mathbf{H}^{(0)}$ to pose-correlated features $\mathbf{H}^{(3)}$. $\mathbf{H}^{(3)}$ is then fed to pose regressor to obtain a preliminary pose result $\mathbf{S}_0$. This pose regressor is composed of a global linear regressor $\mathbf{W}^G$ and K exemplar based nonlinear regressor.	36
3.3	Some sampled exemplars, each of which represents a distinct pose-appearance pair. Note that the actual exemplar's appearance features are the embedded features at the third layer, here we show their corresponding facial images.	38

- 3.4 The architecture of a Localized Deep Auto-encoder Network (LDAN).
 In the first layer, the i th pose-indexed patch (pose-indexed patch is a local patch extracted from the corresponding facial landmark) is encoded by a Local Auto-Encoder (LAE) $\mathbf{W}_i^{(1)}$, and the output is $\mathbf{h}_i^{(1)}$. The output of LAEs proceed in two paths. On one hand, each $\mathbf{h}_i^{(1)}$ is directly fed into a local regressor $\mathbf{R}_i$ to predict the corresponding landmark's position deviation $\Delta \mathbf{p}_i$. On the other hand, these outputs are concatenated into a single global feature vector $\mathbf{H}^{(1)}$ which is taken as the input for the second layer. Therefore, the global facial structure can be implicitly preserved. Finally, the output $\mathbf{H}^{(3)}$ of the top layer is linked to all local regressors, where both global facial structural information and local landmark feature information are jointly exploited for facial landmark detection. 43
- 3.5 For the right face image: the blue point is the current estimated position of a cheek landmark, and the green point is its ground-truth position. The patch in the left-hand side is its enlarged view, which shows the insufficiency of a single patch on dealing with ambiguous landmark. Intuitively, based on this patch alone, the local regressor might adjust the current estimated position to its nearest edge point (the direction is marked in red arrow to the red point). However, its ground-truth position (green dot) is far away from the estimated position (red point). Thus when the position of a facial landmark is uncertain from a local patch perspective, landmark detection using that local patch alone cannot work well. (Best view in the color pdf file.) 44

3.6	From top to the bottom: the weight visualization of the 49th [3] LAE which corresponds to the left outer corner of the mouth; the 4th LAE which corresponds to the left cheek; the 40th LAE which is the left eye corner; and the 9th LAE which is the chin landmark.	45
3.7	The overall NRMSE distributions of GEDAN, GSDAN and Global-SAN across LFPW, HELEN and IBUG testing datasets.	50
3.8	CED curve of a single GEDAN on the LFPW, HELEN and IBUG datasets, respectively.	51
3.9	The structures of the three baselines and our LDAN. Starting from the top-left one in clockwise order: Semi-LDAN, Ext-LDAN, LDAN and Local-SAN.	56
3.10	CED curve of a single LDAN on the LFPW, HELEN and IBUG datasets, respectively.	57
3.11	The performance gains brought by cascading LDANs, where “G+1LDAN” represents one GEDAN followed by one LDAN, and so on.	59
3.12	CED curves of various face alignment methods evaluated on 66 facial points.	60
3.13	CED curves of various face alignment methods evaluated on 49 facial points.	61
3.14	From the top row to the bottom row are face alignment results on HELEN, LFPW, IBUG respectively. Note that pose, expression and illumination variations are larger in IBUG than the other two datasets. The sixth column enclosed in red rectangle shows three failed alignments. . .	62

4.1	A group selfie photo taken in the real-world scenario. The blue points on the faces are the facial landmark detection results obtained by our approach. Note that faces are rotated in a wide range of angles. Moreover, some facial images are blurry and partially occluded.	65
4.2	The pipeline of our HiCNN for facial landmark detection.	66
4.3	The architecture of our Global CNN. Our Global CNN consists of three sub networks, namely the Sub-NET, the Sel-NET and the Top-NET. The Sub-NET forms the basis for both the Sel-NET and Top-NET. Letters “C, P, R, S” embedded in the arrows represent a convolutional layer, a max pooling layer, an Rectified Linear Units (ReLU) layer [4] and a softmax layer respectively. <i>e.g.</i> , “P,R” denote a max pooling layer followed by an ReLU layer. Each round-corner blue rectangle denotes a feature map group, whose size is specified at its bottom in a form of $w \times h \times c \times N$, where w and h indicates the width and height of a feature map, c and N denotes the number of channels and number of instances in the feature map group. The images on the left-hand side of Sub-NET are some examples of our training data, which span a full range of rotation.	69
4.4	The four landmarks we choose to detect by the global CNN are marked as blue asterisks, which are the 38th, 45th, 30th and the average of the 63rd and 67th landmark. The green, black, blue and red dashed rectangles denote the left-eye-brow part, right-eye-brow part, the nose part, and the mouth part respectively. Note that the actual size of the nose part and the size of mouth part are the same, here we enlarge the red rectangle a bit so as to differentiate it from the blue rectangle for clarity.	74

4.5	A partial view of our part-based CNN architecture corresponding with the mouth part. The convolutional layer, max-pooling layer and the ReLU layer are represented by “C,P,R” respectively. “D” denotes the detector layer and “E” is an elementwise product operator.	76
4.6	The structure of our patch-based CNN. The facial patches on the bottom row are pose-indexed patches. The filters of convolutional layers are indicated by various patterned square blocks, where square blocks of the same color and pattern represent the same filters. The blue rounded rectangles denote feature maps, whose size are indicated by $w \times h \times c$	79
4.7	Some example samples for the global CNN with their corresponding landmark positions marked as red dots. The samples in the blue rectangle are augmented training images, and the ones in the red rectangle are randomly transformed LFPW testing images.	83
4.8	The NRMSE results of our global CNN against 9 rotation groups on LFPW testing images.	84
4.9	The NRMSE results of comparing approaches on the augmented LFPW testing images.	86
4.10	From the top row to the bottom row are the part images corresponding to facial contour, left-eye, right-eye, nose and the mouth part respectively. Each part image is of the size 61×61	88
4.11	The CED curves of three patch-based CNN models (built from 3 part-based CNN results) on the HELEN testing dataset.	90
4.12	The CED curves of three part-based CNN models on the IBUG testing dataset.	91

4.13	Each row corresponds to a sample mouth part. For each row, the image on the left is the original part image; the central one is the weighted part image; and the right one is the corresponding weight map generated from the detector layer.	92
4.14	The CED curves of comparing models on HELEN testing images. . . .	93
4.15	The filters of the first convolutional layer in three comparing patch-based CNN models. Each row of filters correspond to a landmark or several landmarks, whose indexes [3] are on the left of the filters. The dashed red rectangle encloses the landmark-specific filters and the blue one contains the group-shared filters. Model names are indicated at the bottom of the corresponding filters.	95
4.16	CED curves of face alignment methods evaluated on 68 facial points with ideal bounding boxes.	96
4.17	CED curves of face alignment methods evaluated on 49 facial points with ideal bounding boxes.	97
4.18	CED curves of face alignment methods evaluated on 68 facial points with challenging bounding boxes.	99
4.19	CED curves of various face alignment methods evaluated on 68 facial points with arbitrary rotation.	100
4.20	Some example facial landmark detection results on IBUG dataset. The first two row of images are facial images cropped from challenging bounding boxes, and the bottom two rows are images with arbitrary random rotation angles coupled with random scale as well as translation. . .	102
4.21	Some example failure facial landmark detection results on IBUG dataset.	102

5.1	Several partial face examples. (a) Three partial face patches (in the red ellipse) are from the LFW dataset [1] which are occluded by heads; (b) Face occluded by sunglasses. (c) Arbitrary partial face patch.	105
5.2	Our proposed partial face alignment framework. (a) Feature extraction: keypoints detected by SIFT keypoint detector are marked out as green dots on both images. The left image is the probe partial face image, and the right one is the gallery face image. (b) Keypoint selection by Lowe's matching scheme: roughly matched keypoints of these two images are connected by green lines, while two pairs of impostor matches are linked by red lines. (c) Point set matching procedure: point set of the probe image marked out as blue diamonds are iteratively aligned to the red-marked point set of gallery image by using RPSM. (d) Matching result: the left one is the warped image using the transformation parameters derived from the matching process, and the right one is the gallery image. Through RPSM, the probe partial face is successfully aligned to the gallery image.	106

5.3	Illustration of an RPSM matching process. RPSM successfully aligns the input probe image (bottom left) to the gallery image (bottom center), and the result is a warped probe image (bottom right). Six iterations of this RPSM matching process are depicted in the upper red rectangle, where each iteration is enclosed in a light blue square with its corresponding iteration number It and λ_3 indicated at the bottom. Within each iteration, the blue diamonds represent probe keypoints and red crosses represent gallery keypoints. As the matching process proceeds, probe keypoints are gradually aligned to gallery keypoints. Finally, they're perfectly matched in Iteration 12.	117
5.4	ROC curves corresponding to various values of λ_1 . The topmost red curve corresponds to $\lambda_1 = 0.01$, and the bottom black one belongs to $\lambda_1 = 0.1$	119
5.5	ROC curves corresponding to various values of λ_3 . The bottom red curve corresponds to $\lambda_3 = 0.001$, and the bottom green one belongs to $\lambda_3 = 0.01$	121
5.6	Five example facial pairs generated from AR dataset with their landmarks annotated in red dots. The images in the first row are holistic faces with neutral expression and illumination, and the ones in the second row are their corresponding partial faces.	124
5.7	The CED curves of various comparing approaches on 600 partial faces.	128
5.8	Example facial images from the LFW dataset after random transformation in View 2. Columns in the red box: matched image pairs. Columns in the blue box: mismatched image pairs.	129
5.9	Verification ROC curves of the comparing algorithms on LFW dataset. .	131

5.10	Verification ROC curves of the comparing algorithms on PubFig dataset.	133
5.11	Samples of our selected PubFig dataset. Top row: original selected images of Adriana Lima. Bottom row: corresponding random transformed images of the top row.	134
5.12	Recognition rates of the comparing algorithms at various ranks on PubFig	135
5.13	Samples from the AR dataset. (a) Cropped gallery image with neutral expression. (b) Cropped probe images occluded by sunglasses and scarf. (c) Original sized probe image with occlusion.	136
5.14	Sample probe images in EYB dataset with random block occlusion. Their corresponding occlusion levels are listed underneath.	138
5.15	Sample example faces with various expressions from AR dataset. Facial images in the same column belong to the same identity. The images in the top row are samples from the gallery, and ones in the second and the third row are samples with smiling faces and screaming faces respectively.	140
5.16	Sample faces with various degrees of partialness from the LFW dataset. Facial images in the first column are the gallery images, and the ones from the second to the 7th column are the probe images with increasing degrees of partialness. The numbers s , θ , and t at the bottom of each column (except the first column) are the corresponding transformation parameters. The images in the top row belong to the same identity, and they form a positive match. The gallery face and the probe faces at the bottom row are from two different identities, and they constitute an impostor match.	141
5.17	The ROC curves of RPSM and HDLBP working on 6 groups of LFW View 2 image sets.	142

5.18	Eighteen face alignment examples by RPSM from AR dataset. Each row contains four pair of faces, which are two face alignment examples. Each example consists of two pairs of faces, where the left pair comprises the unaligned partial face and its corresponding holistic gallery image, whose local features chosen by the keypoint selection process are matched by green lines. Correspondingly, the face pair on the right shows the alignment result.	147
------	--	-----

List of Tables

3.1	The difference between CDAN and CFAN.	48
3.2	The overall NRMSE of the GEDAN with various number of exemplars (K).	53
3.3	Pitch, yaw and roll angle distributions of the 689 testing images	54
3.4	NRMSE result (in percentage) of GEDAN and its improvement against the Global-SAN in various rotation groups	54
4.1	The detailed structure of a CNN network in the part-based CNN. “C,P,R,D” represent a convolutional layer, a max pooling layer, an ReLU layer and a detector layer respectively. The variable L in the last row is the num- ber of landmarks to detect in this facial part. For instance, the mouth part has 20 landmarks and the nose part has 9 landmarks.	75
4.2	The overall NRMSE of the global CNN with various number of regres- sion channels (K).	86
4.3	The detailed structure of <i>slim CNN</i> ’s Top-NET and wide CNN’s Top- NET. “C,R” represent a convolutional layer, and an ReLU layer respec- tively.	87
4.4	Comparison of averaged errors with state-of-the-art methods.	98
5.1	Classification accuracy corresponding to various values of λ_1	119
5.2	Classification accuracy corresponding to various values of λ_3	120
5.3	Face verification accuracy corresponding with various distance metrics .	125
5.4	Verification accuracy and corresponding standard deviations of the com- paring algorithms on LFW	132
5.5	Verification accuracy and corresponding standard deviations of the com- paring algorithms on PubFig	132

5.6	Average recognition accuracy (%) of the comparing algorithms at various ranks on PubFig	135
5.7	Recognition accuracy (%) of the comparing algorithms on aligned AR dataset.	138
5.8	Recognition accuracy (%) of the comparing algorithms on original AR dataset.	138
5.9	Recognition accuracy (%) of various algorithms on EYB with different occlusion levels.	139
5.10	Recognition accuracy (%) of the comparing algorithms across facial expressions.	141
5.11	Verification accuracy and corresponding standard deviations of RPSM and HDLBP on LFW.	143

List of Abbreviations

CDAN	Cascaded Deep Auto-encoder Networks
GEDAN	Global Exemplar-based Deep Auto-encoder Network
LDAN	Localized Deep Auto-encoder Network
LAE	Localized Auto-Encoder
CFAN	Coarse-to-Fine Auto-encoder Network
SDM	Supervised Descent Method
GMM	Gaussian mixture model
HiCNN	Hierarchical Convolutional Neural Networks
RPSM	Robust Point Set Matching
MLERPM	Metric Learned Extended Robust Point set Matching
CPD	Coherent Point Drift
GMMreg	GMM registration
CED	Cumulative Error Distribution
NRMSE	Normalized Root Mean Squared Error
LBP	Local Binary Patterns
SIFT	Scale-invariant feature transform

List of Notations

λ	Parameter
x	Single element
$\boldsymbol{x}$	Vector
$\mathbf{X}$	Matrix
$\ \mathbf{X}\ ^2$	The Frobenius norm of $\mathbf{X}$ defined as $\ \mathbf{X}\ ^2 = \sqrt{(\sum_{ij} \mathbf{X}_{ij}^2)}$
$()^T$	Matrix transpose
$\text{tr}(\mathbf{X})$	The trace of $\mathbf{X}$
$\mathbf{Z} = \mathbf{X} \odot \mathbf{Y}$	Elementwise product, <i>i.e.</i> , $\mathbf{Z}_{ij} = \mathbf{X}_{ij} \mathbf{Y}_{ij}$

Chapter 1

Introduction

1.1 Background

With the popularity of social networks and the pervasiveness of smart phones with digital cameras, the Internet is witnessing large amount of photos generated by numerous users every day. People record their daily life with cameras and share their moments with friends instantly. Many of these photos are selfies or group photos which contain faces. From the user's standpoint, it's desirable to have a photo management software, which is able to detect faces from images and group photos by identity. From the social networks' (*e.g.*, Facebook, Renren, *etc.*) viewpoint, automatically conducting facial analysis, such as age estimation, facial expression recognition, and face identification, allows them to gather multifarious information from their users, and redistribute advertisements accordingly.

Attracted by the prospect and benefit of successful face analysis, numerous research efforts have been devoted to this area for several decades. Within the field of face analysis, a major research topic is face recognition. Face recognition aims to identify a target face in the photo. For example, law enforcement officers would like to identify a suspect in the surveillance video from a large face database to initiate apprehension. This

Figure 1.1: Partial face examples. (a) Three partial face patches (in the red ellipse) are from the LFW dataset [1] which are occluded by heads; (b) Face occluded by sunglasses. (c) Arbitrary partial face patch.

process normally consists of two preprocessing steps, namely face detection and face alignment. Face detection scans through the frames and quickly detects facial regions, and face alignment aligns the detected faces to canonical position, which reduces the variability of faces to facilitate the subsequent recognition stage.

While considerable progress has been made in face alignment and face identification, they have difficulty in dealing with facial images taken from in-the-wild scenarios. Specifically, when a facial image is captured, one or multiple of the following scenarios may occur: the facial image can be taken under poorly illuminated environment; the person might be with exaggerated facial expression; the face may be occluded by other faces or objects; the head may be with large poses rotated out-of-plane or in-plane from the frontal pose; the facial image could be cropped due to the limited camera view or inaccurate bounding box generated by face detection, and so on. Faced with these challenges, the face detection results are likely to be inconsistent, where the detected faces may have large scale variations and they can be off-centered from bounding boxes. Based on face detection results, face alignment can fall short of accuracy. As a chain reaction effect, the face recognition performance is drastically compromised.

As mentioned above, facial images can be partially occluded and cropped. In this thesis, these facial images are termed as partial faces [5]. Some example partial face images are shown in Fig. 1.1. Accordingly, we have holistic facial images where most facial components are visible. In this thesis, we also use the term *holistic (partial) face*

Figure 1.2: The system flow of a standard holistic/partial face recognition approach. In this thesis, we focus on the problems of holistic face alignment, partial face alignment and partial face recognition. These three topics are accentuated in blue color.

alignment, which aligns holistic (partial) faces to the canonical template. Similarly, we adopt the term *holistic (partial) face recognition*, which identifies holistic (partial) faces. Fig. 1.2 illustrates the system flow of a standard holistic/partial face recognition approach, where holistic/partial face alignment serves an indispensable role in achieving accurate face identification.

The mainstream holistic face alignment approaches can be grouped in two categories. One group of methods [6, 7, 8, 9] align the face to a predefined pose by detecting the positions of fiducial landmarks and utilizing the affine transformation. This category is also known as facial landmark detection¹ [10]. The other is to align two or more facial images simultaneously by maximizing their appearance similarity on corresponding regions [2, 11, 12]. Fig. 1.3 displays face alignment results of these two groups of approaches. Both categories of methods have achieved promising performance in practice. However, their alignment accuracy degrades greatly when confront the real-world

¹Note that in the literature, facial landmark detection is also termed as *face alignment*, *facial pose regression*, *facial point detection*.

Figure 1.3: Two holistic face alignment examples. (a) Facial landmark detection result achieved by our CDAN, where the green dots represent the detected 68 facial landmarks. (b) Appearance similarity maximization based face alignment result, where facial images in the first row are the unaligned faces with corruption and the faces in the second row are the aligned faces. Facial images of (b) are referenced from the paper RASL [2].

challenging cases. Specifically, most facial landmark detection approaches assume the bounding box provided by the face detection step is consistent, and their performance will drop drastically once bounding boxes drift. Moreover, the alignment accuracy is unsatisfactory on challenging facial poses (*e.g.*, near-profile pose, faces with large in-plane rotation). The appearance similarity maximization based approaches assume that the matching images are roughly aligned at the beginning. This assumption limits their applications in real-world scenarios, since faces might be rotated or translated to some extent.

For the partial face alignment scenario, due to occlusion and random crop, some key facial components can be missing, which poses tremendous challenge to the existing face alignment approaches. Therefore, for the real-world scenarios, the state-of-the-art methods are difficult to meet the need of practical facial analysis applications, and there is a strong demand of robust face alignment approaches for both holistic and partial face images.

Assuming the face alignment step is successful, the mainstream face recognition approaches proceed by constructing generative/discriminative subspace/manifold [13, 14, 15, 16, 17], or by sparse dictionary learning [18, 19, 20], or with deep learning [21, 22, 23]. These face recognition approaches have achieved significant progress and the current state-of-the-art methods [24, 25] have surpassed human-level recognition ability

on the challenging LFW dataset [1]. Nevertheless, most of them are designed for holistic faces. As mentioned previously, human faces might be occluded by other objects in the real-world scenarios, especially in unconstrained environments such as smart visual surveillance systems. Hence, we have to recognize the person of interest from his/her partial faces. To deal with face occlusions, various algorithms based on sparse representation have been proposed recently [18, 19, 20, 26, 27]. [20] was the pioneer work in this area, where sparse representation was utilized to reconstruct occluded or stained facial images and to align probe face images to gallery images. While these approaches can achieve encouraging recognition performance in case of occlusions, they may fall short if the probe image is an arbitrary face patch. Hence, it's of interest to devise a partial face recognition approach to robustly identify arbitrary facial patches.

To summarize, for the current face alignment and recognition literature:

- A more robust and accurate holistic face alignment approach is desirable. Specifically, it shall be robust in dealing with large facial poses and inconsistent face detection to allow for automatic holistic face recognition.
- Partial face alignment and partial face recognition are of great interest in dealing with faces with severe occlusion and arbitrary facial patches, *i.e.*, it shall be able to align a probe partial face to a gallery facial image, and measure their similarity accordingly.

1.2 Motivation

Generally speaking, for the holistic face alignment, face landmark detection approaches are more favorable than the methods based on appearance similarity maximization. This is because the former can lead to various applications other than face alignment. For example, by analyzing the movement of facial landmarks, facial expression analysis can be performed. Hence, in this thesis, we focus on advancing face landmark detection

for the holistic face alignment problem. The mainstream facial landmark detection approaches proceed with two steps: pose initialization and pose update. The pose initialization stage generates an initial face shape (*i.e.*, the facial landmark positions) from a facial image, and the pose update stage adjusts the face shape progressively. A group of methods [9, 28, 29, 30, 31, 32] employ mean shape as the starting pose. This strategy depends heavily on the face detection accuracy, and it works poorly when testing faces have large rotation range or scale variations. To overcome this problem, Burgos-Artizzu *et al.* [33] and Cao *et al.* [6] utilized multiple random shapes as initializations and used the median regression result as the final pose. While one of these random initial guesses might strike the ground truth, the median result may divert as it relies on the overall regression outputs. Alternatively, Zhang *et al.* [34] and Sun *et al.* [35] deployed a Stacked Auto-encoder Network (SAN) and a CNN respectively to generate the initial pose from a low-resolution facial image. Their works greatly increase the alignment robustness against large pose variation, which shows a promising direction in initializing facial pose. However, regressing on holistic images poses a great difficulty for these conventional deep models. By tailoring deep models for landmark detection, further improvement on pose initialization can be achieved. Specifically, we have designed the GEDAN model (a component network in the CDAN) and the global CNN (a network in the HiCNN), which are two networks transformed from conventional auto-encoder network and convolutional neural networks respectively. Both of them achieve good robustness and high accuracy on pose initialization.

For the pose update stage, various face alignment approaches [6, 33, 34, 36, 37] regress on pose-indexed features with a feature-to-pose mapping function. Several boosted regression approaches [6, 36, 37] randomly generate pose-indexed ferns as local features to achieve fast detection speed. The local ferns are extracted from sparse locations in the neighborhood of the facial landmarks, which are arguably weak to represent local patches. More recently, both Zhang *et al.* [34] and Xiong *et al.* [8] utilized

pose-indexed SIFT features extracted from pose-indexed patches as feature representations. SIFT is hand-crafted feature, it's desirable to learn feature extractors from the data directly. Hence, for the pose update stage, a possible direction of increasing alignment accuracy is to learn pose-informative features and feature-to-pose mapping function in a unified data-driven way.

For the partial face alignment and partial face recognition, since the face might be severely occluded or cropped, it's challenging to estimate facial landmark positions. This is because some key components (*e.g.*, mouth, eye) can be missing from a partial face, which undermines the facial structure and increases the facial appearance variance. While detecting landmark positions from partial faces is difficult, we observe that there exist some similar local facial patches between two partial faces. For example, two partial faces might share a common visible nose region. By localizing these semantic alike facial patches from both faces, we are able to align them to each other. In other words, we can build correspondence between facial patches of the matching partial faces. Subsequently, the similarity of two partial faces can be formulated by calculating the likeness between these corresponding facial regions and the area of matching facial patches. For instance, if two partial faces share a large near-identical facial patch, it'll be highly probable that they are from the same identity. Thus, partial face alignment and recognition can be achieved by localizing semantic alike regions from the matching partial faces, and subsequently calculating the similarity between these corresponding patches.

1.3 Main Contributions of the Thesis

Inspired by the aforementioned ideas, we propose two branches of works. Specifically, for the holistic face alignment, we present two deep learning based approaches to deal with large facial poses and inconsistent face detection. We further present a partial face

alignment and recognition method which automatically aligns a partial face to its corresponding gallery face, and subsequently recognizes its identity. The main contributions of the thesis are summarized as follows:

First, to increase the alignment accuracy against large facial poses, we propose a Cascaded Deep Auto-encoder Networks (CDAN) approach for holistic face alignment. Our CDAN consists of two new auto-encoder network structures, namely the Global Exemplar-based Deep Auto-encoder Network (GEDAN) and the Localized Deep Auto-encoder Networks (LDAN). In particular, GEDAN’s top regression layer deploys several exemplars to assist pose estimation from holistic facial image, which improves robustness against large facial poses compared to the conventional auto-encoder network. In terms of LDAN, its first layer consists of individual Local Auto-Encoders (LAEs) to extract pose-informative features from its corresponding local patches. The outputs of these LAEs are then directly fed into their corresponding local regressors. In addition, these outputs are concatenated into a global feature vector which is further encoded by several layers of auto-encoders to preserve the global facial structure. Thus, pose-informative features and feature-to-pose mapping are learned in a unified way. Our CDAN achieves superior alignment accuracy with real-time speed on three public testing benchmarks.

Second, to further advance the frontier of holistic face alignment, we leverage the strength of the Convolutional Neural Networks (CNN) and devise a Hierarchical CNN (HiCNN) cascade. Specifically HiCNN consists of a global CNN, a part-based CNN and a patch-based CNN. For the global CNN, we design a multi-channel switch which splits the facial images into multiple groups, where facial images in each group have similar poses and they are fed to a corresponding regression channel. Each channel therefore only deals with a small range of scale variations and rotation, and the face alignment difficulty is substantially mitigated. For the part-based CNN, we add a *detector* layer to identify and suppress the occluded and uninformative facial region, thereby increasing

alignment robustness against occlusion. For the patch-based CNN, we exploit the fact that neighboring landmarks share similar facial structure, and devise a local partially sharable filter structure for each landmark. The features from these local filters are then fused together at the top fully-connected layers to jointly regress the landmark positions. The proposed HiCNN is able to accurately detect landmarks from facial images with *arbitrary* in-plane rotation, large scale variations and random face shifts.

Third, in dealing with partial faces, we propose a partial face alignment and recognition approach named Robust Point Set Matching (RPSM). RPSM casts the problem of face alignment and recognition as the one of matching two local feature sets. Through matching, partial probe face is aligned to the gallery holistic image, and the similarity of two faces is converted as the distance between two feature sets. The matching problem is formulated in a linear programming framework, where constraint of affine transformation can be easily applied. Extensive experiments on four face datasets verify the effectiveness of the proposed RPSM both on partial face alignment and partial face recognition.

1.4 Organization of the Thesis

This thesis is organized as follows:

In Chapter 2, we first describe the face alignment methods, and then we review relevant face recognition related approaches.

In Chapter 3, our proposed Cascaded Deep Auto-encoder Networks (CDAN) approach for holistic face alignment is introduced, and extensive experiments are presented.

Chapter 4 describes our Hierarchical CNN (HiCNN) cascade for holistic face alignment. A thorough comparison with state of the arts on three benchmarks is given. Moreover, face alignment performance with challenging bounding boxes and on arbitrary

rotated facial images are investigated.

In Chapter 5, we detail the proposed RPSM approach, which is capable of performing partial face alignment and recognition robustly. Comprehensive experiments on partial face alignment and partial face recognition on four benchmarks are conducted to validate its effectiveness.

Chapter 6 presents the concluding remarks on this thesis and recommends several future research directions.

Chapter 2

Related Works

This chapter includes a survey of important approaches related to the proposed methods in this thesis. In Section 2.1, a general review of face alignment works is introduced in details. In Section 2.2, relevant works of partial face recognition are reviewed.

2.1 Face Alignment

Face alignment is a crucial step in a face recognition system, and it plays an important role in achieving high face recognition rate [21, 38]. The face alignment process aligns a detected face to the canonical position, which reduces the variability of faces to facilitate the following recognition stage. Since most of the existing face alignment approaches focus on aligning holistic facial images, unless otherwise stated, *face alignment* in this section represents *holistic face alignment*. Face alignment methods can be categorized into two groups, namely face alignment based on facial landmark detection and the one based on appearance similarity maximization. Facial landmark detection methods [10] detect semantic facial landmarks, such as mouth corners, nose tip, chin *etc.*, and utilize affine or similarity transform to normalize the face to the frontal pose. For appearance similarity maximization, it align a batch of images simultaneously such that the mutual appearance similarity between the transformed images can be maximized [2, 11, 12]. In

the rest of this section, a review of face alignment approaches from both of these two groups is provided, which is followed by a discussion on how to apply these methods on partial faces.

2.1.1 Face Landmark Detection

The mainstream face landmark detection approaches can be mainly divided into four groups, discriminative fitting [7, 30, 32, 39, 40, 41, 42, 43, 44], boosted regression [8, 33, 36, 37, 45, 46, 47], deep learning based methods [34, 35, 48, 49, 50], and occlusion-robust approaches [33, 51, 52].

2.1.1.1 Discriminative fitting

Discriminative fitting approaches predict facial landmarks by maximizing the joint posterior probability over all landmarks given the input image. Generative model fitting such as Active Shape Models (ASMs) and Active Appearance Models (AAMs) [9, 28, 53, 54, 55, 56, 57, 58, 59, 60] are classic face alignment models. They learn a shape and appearance variation model using PCA and deploy a complicated iterative update scheme to adjust the model parameters. Specifically, for an AAM model, its shape model is defined as

$$\mathbf{s} = \mathbf{s}_0 + \sum_i p_i \mathbf{s}_i, \quad (2.1)$$

where shape $\mathbf{s}$ records the landmark positions, $\mathbf{s}_0$ is the mean shape, and p_i is the shape parameter. Similarly, the appearance model is formulated as

$$\mathbf{A} = \mathbf{A}_0 + \sum_i \lambda_i \mathbf{A}_i, \quad (2.2)$$

where $\mathbf{A}$ is the appearance vector, and λ_i is its corresponding appearance parameter. To fit an AAM model to an unseen image, the shape coefficients and appearance parameters

are iteratively updated in a Lucas-Kanade form [28]. A recent work [43] shows that when trained in the wild, generative models can perform very accurate model fitting.

Constrained Local Models (CLMs) [30, 32, 42, 61] use generative deformable part models where each part is associated with a local template detector. For each testing sample, the model iteratively optimizes parameters to maximize the sum of responses generated from local classifiers. To speed up the fitting procedure, most of the existing CLM methods approximate the response maps using a parametric distribution model (*e.g.*, Gaussian distribution, mixed Gaussian distribution Model) or non-parametric model [31]. However, these approximations bring in fitting inaccuracy. Differently, [39, 40] fit the response maps of local classifiers by selecting from a pool of training exemplar poses. Nevertheless, the speed of this type of approaches are usually slow, which limits their applicability in real-world scenarios.

Both the AAMs and CLMs utilize the mean shape model to initialize the pose estimation, which compromises their estimation robustness in dealing with large facial poses. Differently, both our CDAN and HiCNN employ deep models to provide a preliminary starting landmark configuration; thus following local adjustment is able to perform delicate pose refinement.

2.1.1.2 Boosted regression

Boosted regression approaches [6, 8, 36, 37, 62, 63, 64, 65] learn a cascade of feature-to-pose mappings to address the highly nonlinear pose estimation problem. For example, Cao *et al.* [6] presented a Cascaded Pose Regression (CPR) model for face alignment, where a series of weak regressors based on pose-indexed random ferns were trained to gradually refine the initialized pose. Specifically, CPR adopts boosted regression to assemble multiple weak regressors $R_1, R_2, \dots, R_T$ together in a cascaded manner. Each weak regressor derives a pose derivative δs from pose-indexed features and then revises

$\mathbf{s}$, *e.g.*, R_t adjusts $\mathbf{s}_{t-1}$ as follows,

$$\mathbf{s}_t = \mathbf{s}_{t-1} + R_t(I, \mathbf{s}_{t-1}), \quad t = 1, \dots, T \quad (2.3)$$

where I is the facial image to be aligned. Let's denote the ground-truth pose of the i th sample as $\hat{\mathbf{s}}_i$, the regressors are learned such that the training error is minimized

$$R_t = \operatorname{argmin}_{R_t} \sum_i \|\hat{\mathbf{s}}_i - (\mathbf{s}_{t-1} + R_t(I, \mathbf{s}_{t-1}))\|^2. \quad (2.4)$$

One important factor for the success of CPR is the usage of pose-indexed features which are features extracted based on the current facial landmark positions. In particular, CPR employs pose-indexed pixel-difference features, *i.e.*, the intensity contrast of two pixels indexed by a facial landmark. This strategy makes the feature adaptive to facial poses, thereby increasing CPR's robustness against in-plane rotation and translation.

Based on this framework, Kazemi *et al.* [36] proposed an ensemble of regression trees learned from gradient boosting to achieve high quality predictions. Xiong *et al.* [8] devised a Supervised Descent Method (SDM) for face alignment, where pose-indexed SIFT features are adopted and the weak regressor is cast as a linear projection matrix. Similarly, Ren *et al.* [37] utilized random ferns to learn the pose-indexed features and derived a linear function to model the feature-to-pose mapping. This approach achieved extremely fast speed, reaching 3000 FPS. While these regression methods achieve fast alignment speed, their alignment accuracies in the wild are still unsatisfactory. This is because most of them adopt mean shape as the initial shape for face alignment. If the target image is a near-profile face or the bounding box is drifted from the ground-truth face region, the alignment process can be easily trapped in local minimum. Recently, Zhu *et al.* [46] devised a coarse-to-fine shape searching approach to gradually narrow down the possible shape space in which the testing face resides, thereby effectively circumventing the pose initialization problem. It's yet unclear how their approach can

be generalized to cope with arbitrary in-plane face rotation while maintaining the real-time speed.

One advantage of this group of approaches is their fast speed, which is mainly achieved by the simple pose update scheme, *i.e.*, linear feature-to-pose mapping. While the linearity reduces the pose estimation workload, it sacrifices the capability of the model. This inadequacy becomes more evident when the initialized pose is far from the ground truth. To mitigate the problem, more complex feature-to-pose mapping scheme can be employed. Deep neural network is one promising option, and the approaches based on this category are described in the following section.

2.1.1.3 Deep learning based approaches

Deep learning approaches [34, 35, 49, 66] utilize deep neural networks to estimate landmark positions. Zhang *et al.* [34] presented a Coarse-to-fine Stacked Auto-encoder Networks (CFAN) method for face alignment. CFAN employs the global stacked auto-encoder network (Global-SAN) to obtain a preliminary shape from the low-resolution facial image for the pose initialization. The subsequent local auto-encoder networks (Local-SAN) gradually adjust the facial pose by regressing on the pose-indexed SIFT features with higher image resolution. In particular, the Global-SAN deals with the holistic input image directly and learns a nonlinear function $f_G(I)$ to estimate the landmark positions $[p_1, p_2, \dots, p_M]$, where I represents the raw pixel features of the image and M is the number of landmarks. Based on this preliminary result, pose-indexed SIFT features are extracted from facial patches surrounding the facial landmarks. Specifically, SIFT features are derived on each facial patch and they are projected to low dimension by PCA. This low-dim features are then fed to the Local-SAN to obtain the local landmark position adjustment $[\Delta p_1, \Delta p_2, \dots, \Delta p_M]$. For the i th landmark, its position is updated by $p_i = p_i + \Delta p_i$. Fig. 2.1 shows the structure of the Global-SAN and the one of Local-SAN.

Figure 2.1: The Global-SAN and the Local-SAN employed in the CFAN model.

Sun *et al.* [35] presented a multi-stage Convolutional Neural Networks (CNN) for landmark detection. Specifically, a CNN is deployed to initialize the landmark positions and another two networks are utilized to locally refine the shape. Similarly, Zhou *et al.* [67] devised a four-level CNN cascade for face alignment. The first two-level CNNs estimate landmarks from the holistic facial image, and the next two-level CNNs refine the landmark positions from part (component) images.

Note that all these deep models share similar coarse-to-fine strategies. This is because deep neural networks have much more parameters than shallow approaches such as cascaded regression. It's thereby desirable to get a preliminary facial pose from the low-resolution image at the beginning, and progressively refine the pose on pose-indexed features extracted from higher image resolution. This strategy substantially decreases the computational load compared to applying deep models directly on the high-resolution image. Appealed by this benefit, we also adopt the coarse-to-fine framework in our proposed CDAN (Chapter 3) and HiCNN (Chapter 4) approach. The main difference of our methods with the existing deep learning based models lies at the specific design of neural networks, which will be detailed in the respective chapters.

2.1.1.4 Occlusion-robust approaches

There exist a few face alignment works dealing with facial occlusion explicitly [33, 51, 52]. Robust Cascaded Pose Regression (RCPR) [33] incorporates the occlusion status of facial landmarks into the training label, and utilizes cascaded regression framework to jointly obtain the landmark position as well as the occlusion status. However, this work requires extensive labor on collecting occluded facial images and annotating them with occlusion status. [51] deploys hierarchical deformable part models to estimate landmark positions and augments the training data with numerous synthetically occluded instances. Different from these models, the occlusion detector layer of our HiCNN doesn't require either occlusion labeling or synthetical occlusion data generation.

2.1.2 Appearance Similarity Maximization

Instead of detecting facial landmark positions, appearance similarity maximization based face alignment defines a similarity measure of an image batch and an optimization method to maximize this similarity measure. The mainstream approaches can be grouped into two classes, namely congealing based and low rank based methods.

2.1.2.1 Congealing based methods

The classical work *congealing* [68] employs the sum-of-entropies criterion as the similarity measure, and it updates affine transformation parameters corresponding with each image in a sequential manner. The images are piled together to build pixel stacks, and the entropy of the i th pixel stack is defined as

$$H_i = -\left(\frac{N_0}{N} \log \frac{N_0}{N} + \frac{N_1}{N} \log \frac{N_1}{N}\right), \quad (2.5)$$

where N_0 and N_1 are the number of zeros and ones in the i th pixel stack respectively. The sum-of-entropies can then be represented as $\sum_i H_i$. To minimize this entropy sum,

the affine transformation parameters (*e.g.*, rotation angle, scale, and translation) of the piled images are updated one by one. A particular parameter adjustment would be applied if it decreases the entropy sum.

Huang *et al.* [11] extended this approach to align complex images such as faces by congealing with SIFT descriptors. Cox *et al.* [69] presented a least squares congealing by formulating the objective function in a least square form, such that Gauss-Newton optimization can be adopted.

A major shortcoming of this congealing based works is that they require the target images be roughly aligned at the beginning, which limits their application in the real-world scenarios where faces can be rotated and largely translated.

2.1.2.2 Low rank based methods

Low rank based methods achieve batch alignment by assuming the aligned faces exist in a low-rank subspace. The alignment is performed by minimizing the subspace rank with affine/similarity transformation of the image batch. The early work [70] proposed by Frey and Jojic utilized the expectation maximization algorithm to simultaneously derive principal components and spatial transformations. However, the spatial transformations are limited to draw from a predefined transformation group. Improving on this, Schweitzer [71] presented an alignment method which performs principal component pursuit and transformation parameter learning analytically.

Alternatively, Vadaldi *et al.* [72] devised an alignment method where similarity measure is converted as matrix rank. This measure is not robust since matrix rank can be easily affected by occlusion instead of geometric transformation. To circumvent this problem, Peng *et al.* [2] proposed a more robust method named RASL which takes sparse errors into consideration while minimizing subspace rank. In particular, denoting the original image matrix as D , and image-wise transformation function as τ , the transformed image matrix is then represented as $D \circ \tau$. RASL splits the transformed image

matrix into two terms: one is the low-rank subspace term A and the other is the error term E induced by occlusion. The objective function is formulated as

$$\min_{A,E,\tau} \text{rank}(A) + \gamma \|E\|_0, \text{ s.t. } D \circ \tau = A + E. \quad (2.6)$$

Since this problem is NP-hard, the author relaxed the objective to convex function by replacing the rank function with nuclear norm, and the ℓ_0 -norm by ℓ_1 -norm. The new objective function is given by

$$\min_{A,E,\tau} \|A\|_* + \gamma \|E\|_1, \text{ s.t. } D \circ \tau = A + E. \quad (2.7)$$

The optimal value of τ can be subsequently derived by iterative update.

A recent work [73] combined the strength of both gradient orientation and low rank structure to improve the alignment robustness against occlusion and extreme illuminations.

Compared to the congelating based methods, these low-rank approaches are generally more robust and can align faces with larger pose variations. However, the low-rank based approaches assume the target images to be aligned belong to the same identity. This assumption confines their application to the case where multiple images per person exist. It's unclear how to extend the approaches to align two faces together, *e.g.*, aligning a probe face to the gallery face.

2.1.3 Discussion

For the holistic face alignment, both face landmark detection and appearance similarity maximization achieve promising results. Compared to the appearance similarity maximization, the face landmark detection's result (*i.e.*, facial landmark positions) can lead to various applications other than face alignment. For instance, analysis of landmark movement is an efficient method for facial expression recognition. Another advantage

of landmark detection approaches is their greater robustness against illumination and pose variations. Therefore, in this thesis, we focus on the facial landmark detection problem.

For partial face alignment, it depends on whether the testing facial image is an occluded partial face or an arbitrary partial face patch (refer to Fig. 1.1). Both facial landmark detection and appearance similarity maximization can be readily applied on occluded faces, provided that the occluded partial face is near upright. On the other hand, their performance on arbitrary partial face patches is unsatisfactory. In Chapter 5 we'll investigate their performance on partial face alignment and compare them with the proposed RPSM.

2.2 Partial Face Recognition

Traditional face recognition approaches [15, 20, 74, 75, 76, 77, 78, 79, 80, 81] work well on holistic aligned faces. In practice, however, faces can be occluded by other objects, resulting in partial faces. Hence partial face recognition is quite beneficial for real-world applications. In the literature, most face recognition approaches are dedicated to holistic face identification. For these methods, the readers are referred to the well-written literature survey [82]. On the contrary, only a few works have been proposed to tackle the problem of partial face recognition. In this section, we focus on this partial face recognition problem. Nevertheless, some holistic face recognition approaches can also be applied to recognize occluded partial faces. In the following, we briefly review several existing approaches which are closely related to partial face recognition. Since our partial face recognition approach is based on point set matching, we also introduce multiple relevant methods on point set matching.

2.2.1 Robust Face Recognition

There are a number of face recognition approaches that are closely related to the partial face recognition problem. These methods can be roughly categorized into five groups, namely Sparse Representation based Classification (SRC) methods, part based representation methods, local features based methods, metric learning based methods, and partial face recognition approaches.

2.2.1.1 Sparse representation based classification methods

To deal with occlusions, many Sparse Representation based Classification (SRC) methods have been proposed in recent years [18, 19, 20, 27]. Wright *et al.* [20] proposed the pioneer sparse representation framework to deal with face occlusion and face corruption. Specifically, they cast the reconstruction problem as ℓ_1 -minimization problem. First, the testing samples are constructed as a sparse linear combination of training samples. Given the training samples of the i th identity: $A_i = [v_{i,1}, v_{i,2}, \dots, v_{i,n_i}] \in \mathbb{R}^{m \times n_i}$, the input probe facial image $y \in \mathbb{R}^m$ is represented as

$$y = \alpha_{i,1}v_{i,1} + \alpha_{i,2}v_{i,2} + \dots + \alpha_{i,n_i}v_{i,n_i}, \quad (2.8)$$

Since the identity of the testing sample is unknown, they introduce a new matrix A for the entire training set by concatenation of the n training samples of all k identities:

$$A = [A_1, A_2, \dots, A_k]. \quad (2.9)$$

Then, the representation of y by A is given by

$$y = Ax, \quad (2.10)$$

where $x = [0, \dots, 0, \alpha_{i,1}, \alpha_{i,2}, \dots, \alpha_{i,n}, 0, \dots, 0]^T \in \Re^n$ is a parameter vector whose nonzero coefficients are concentrated to the i th class.

The authors proposed to use ℓ_1 -minimization to solve the above problem:

$$\hat{x} = \arg \min \|x\|_1 \quad s.t. \quad Ax = y. \quad (2.11)$$

With $\hat{x}$ at hand, the authors proceeded to the classification session, where they defined an $\Re^n \mapsto \Re^n$ function $\delta_i(\hat{x})$ to select coefficients from the i th subject. Then y is classified to the identity who minimizes the reconstruction residual:

$$i = \arg \min r_i(y) = \|y - A\delta_i(\hat{x})\|. \quad (2.12)$$

Based on this SRC framework, and motivated by the success of nonnegative matrix factorization [83], Liu *et al.* [27] imposed nonnegative constraints on the sparse coefficients. Instead of pursuing sparsity, Zhang *et al.* [26] presented a Collaborative Representation based Classification (CRC) method which achieves higher recognition accuracy than SRC based approaches especially when the gallery identities have small sample size. Jia *et al.* [19] exploit the fact that occluding patterns are largely structured and introduced structured sparsity-inducing norms.

While these methods have achieved encouraging recognition performance on occluded facial images, they may fail to work well if the probe image is an arbitrary face patch. This is because these approaches usually require the probe image be roughly aligned to the gallery images, which is not the case for the arbitrary face patch. Differently, the our RPSM method doesn't impose such constraint.

2.2.1.2 Part based representation methods

Several part based representation methods have also been proposed for robust face recognition [84, 85, 86, 87, 88, 89], where each face image is divided into many blocks.

The similarity of these small blocks are computed first and subsequently integrated for face matching. Min *et al.* [86] proposed to detect occluded regions first and then describe the non-occluded facial part as block-based LBP. Their method focuses on the occluding face recognition with scarf and sunglasses. Specifically, each image is split into an upper half and a lower half, which are corresponding to the sunglass and scarf region respectively. These two part images are described by Gabor features, which are then fed to SVM classifiers to detect if a particular part is occluded. If a part is detected as non-occluded, it will be divided into dense blocks. Each block is described by an LBP histogram, and the final derived features are the concatenation of these LBP features. Finally, the distance between two faces is cast as the similarity between two concatenated features. However, in real-world scenarios, occlusion patterns are highly unstructured and occlusion detection can be unreliable.

Alternatively, Chen *et al.* [84] built Stringface from line segments and cast the face recognition problem as a string-to-string matching problem. Pan *et al.* [87] utilized LBP [85] to describe face blocks and employed AdaBoosting to select the most discriminative LBP histograms. Classifiers are then trained independently on each face part, and they are fused to derive the final similarity score. Li *et al.* [88] augmented dense local features with feature locations and utilized Gaussian Mixture Model to capture correspondence of matching local feature pairs. Face verification is then achieved by training SVM classifier on concatenation of the difference features. Simonyan *et al.* [90] extracted dense SIFT features from facial blocks and proposed the Fisher vectors to describe the distribution of these SIFT features. Metric learning is applied to learn discriminative features from these Fisher vectors.

Similar to the SRC-based methods, all these part-based methods require a preliminary face alignment to roughly align the probe and gallery faces at the beginning. The partial face alignment problem itself is quite challenging, and a robust face alignment step is equally important as the recognition step. The proposed RPSM consists of a

partial face alignment step and a recognition step. Hence, the face identity can be recognized in an automatic way.

2.2.1.3 Local features based methods

There have been some attempts on face recognition with local features. For example, Bicego *et al.* [91] extracted SIFT features on both the probe and gallery images and devised three matching schemes. Their work assumed that facial images to be matched were roughly registered. Geng and Jiang [92] employed SIFT features for face recognition, where face recognition is considered as a generic object recognition problem by using local feature matching. Nevertheless, the textural and geometric information are matched independently. Moreover, the geometric correspondence is simply achieved by a standard affine transformation. Dreuw *et al.* [93] employed SURF feature representations for face recognition. They found that SURF is more robust to illumination variation than SIFT, and SIFT is more robust to viewpoint variation than SURF. Ahonen *et al.* [85] proposed an LBP feature representation method for face recognition, which achieves excellent face verification performance on LFW when it is extracted from multiple scales. Tan and Triggs [94] generalized LBP to Local Ternary Patterns to deal with the illumination variations problem. Recently, Chen *et al.* [95] proposed a High Dimensional LBP (HDLBP) approach to take advantage of the blessings of high dimensionality. In particular, they extracted LBP features from multi-scale pose-index patches. Pose-indexed patches are patches cropped from the detected facial landmarks. Each patch is divided into 4×4 blocks, and each block is described by LBP features. These LBP features are concatenated into a high dimensional feature vector, which is followed by dimension reduction. Having reduced the dimension, the authors utilized Joint Bayesian [96] method to perform face verification. This HDLBP approach achieves promising recognition results and we will compare with it extensively in our experiments in Chapter 5.

Our RPSM also utilizes local features for recognition. In particular, it extracts SIFT, SURF and LBP features. Compared to the aforementioned approaches, RPSM doesn't assume a preliminary face alignment is achieved. Instead it explicitly performs face alignment. Moreover, it considers both geometric consistency and textural similarity between local feature sets during the matching process.

2.2.1.4 Metric learning based methods

A number of metric learning approaches have been devised for the face recognition problem during the last decade [97, 98, 99, 100, 101, 102, 103]. These approaches seek to derive a robust distance metric such that the within class variance is reduced and the between class variance is enlarged.

Weinberger *et al.* [103, 104] proposed a large margin nearest neighbor (LMNN) classification metric for k-nearest neighbor (kNN) classification. In particular, the LMNN endeavors to pull instances within the same neighborhood sharing the same label (termed as *target neighbors*) together, while keeping neighbors with different identities far away. Let's denote the training samples as $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M$ with their corresponding labels $y_1, y_2, \dots, y_M$. For each sample, it has K nearest target neighbors. The neighborhood relationship is denoted by η_{ij} , where $\eta_{ij} = 1$ represents that $\mathbf{x}_j$ is one of the K target neighbors of $\mathbf{x}_i$. Let's denote the distance metric as $\mathbf{L}$, then the distance between two instances is given by

$$D(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|^2. \quad (2.13)$$

The cost function of LMNN is formulated as follows

$$\min_{\mathbf{L}} \sum_{i,j} \eta_{ij} \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 + c \sum_{ijl} \eta_{ij}(1 - y_{il}) [1 + \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 - \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_l)\|_2^2]_+, \quad (2.14)$$

where the first item pulls instances with their target neighbors close to each other, and the second one pushes away samples with different identities yet lying within the same

neighborhood. Parameter c controls the balance between these two items. $y_{ij} = 1$ indicates $y_i = y_j$, and $[z]_+ = \max(z, 0)$ represents the hinge loss.

Similar with LMNN, Neighborhood Components Analysis (NCA) [105] learns a distance metric to maximize the leave-one-out kNN softmax score on the training set. Davis *et al.* [100] devised an Information-theoretic metric learning (ITML) approach. The objective of ITML is to optimize the distance metric under various constraints and prior knowledge. The optimization is achieved by enforcing that the learned metric be as close as possible to a known prior. Guillaumin *et al.* [101] presented a logistic discriminant based metric learning (LDML) approach. The LDML measures the probability of two instances sharing the same label by using sigmoid function $p_{ij} = \sigma(b - D(\mathbf{x}_i, \mathbf{x}_j))$, where $\sigma(z) = (1 + e^{-z})^{-1}$ and b is the bias. The objective of LDML is then given by

$$L = \min_{\mathbf{L}} \sum_{i,j} y_{ij} \ln p_{ij} + (1 - y_{ij}) \ln(1 - p_{ij}). \quad (2.15)$$

The above-mentioned methods are mainly linear models which learn low-dimensional embeddings. These approaches may not capture the nonlinearity of facial images. To exploit the nonlinearity, Hu *et al.* [98] devised a Discriminative Deep Metric Learning (DDML) method to derive a deep multi-layer perception network, where the distance of positive face pairs is decreased and the one of negative pairs is enlarged.

2.2.1.5 Partial face recognition methods

To our knowledge, only a few seminal works on partial face recognition have been presented [5, 106]. The objective of partial face recognition is to recognize the person from an occluded partial face or an arbitrary partial face patch. [5] was the first attempt on partial face recognition, where each partial face image is represented by local MKD-GTP features. These local features are then sparsely reconstructed by gallery feature set. Finally, the reconstruction error is converted as the similarity metric between two

matching faces. Nonetheless, the geometrical information of local features is ignored in their method.

To robustly match the probe partial face image with a gallery image, our previous work MLERPM [106] considered partial face recognition as a feature set matching problem, where geometric features and textural features were matched simultaneously. Specifically, let $\{\mathbf{l}_1^P, \mathbf{l}_2^P, \dots, \mathbf{l}_{N_P}^P\}$ be the geometric feature set of the probe partial face image, and $\{\mathbf{t}_1^P, \mathbf{t}_2^P, \dots, \mathbf{t}_{N_P}^P\}$ be the corresponding textural features. N_P is the number of probe keypoints, and $\mathbf{l}_i = [x, y]'$ records the geometric location of the i th probe keypoint. Similarly, for a particular gallery image, we have $\{\mathbf{l}_1^G, \mathbf{l}_2^G, \dots, \mathbf{l}_{N_G}^G\}$ and $\{\mathbf{t}_1^G, \mathbf{t}_2^G, \dots, \mathbf{t}_{N_G}^G\}$ as the geometrical and textural feature set respectively. The objective function of MLERPM is formulated as

$$\begin{aligned}
 J = \min_{f, \mathbf{M}} & \sum_{i,j} \mathbf{M}_{ij} \left(\|f(\mathbf{l}_i^P) - \mathbf{l}_j^G\|_2^2 + \lambda_1 \|\mathbf{t}_i^P - \mathbf{t}_j^G\|_{\mathbf{W}}^2 \right) \\
 & - \tau \sum_{i,j} \mathbf{M}_{ij} + C \sum_{i,j} \mathbf{M}_{ij} \log \mathbf{M}_{ij} + \lambda_2 \Psi(f), \\
 s.t. & \sum_{j=1}^{N_G} \mathbf{M}_{ij} \leq 1, \sum_{i=1}^{N_P} \mathbf{M}_{ij} \leq 1, \mathbf{M}_{ij} \geq 0,
 \end{aligned} \tag{2.16}$$

where $\mathbf{M}$ is the correspondence matrix and $\mathbf{M}_{ij}$ denotes the correspondence from the i th keypoint of the probe image to the j th keypoint of the gallery image. $\mathbf{W}$ is the metric matrix corresponding with the textural similarity between two feature points. f is the geometric non-affine transformation function and $\Psi(f)$ calculates its non-affineness. The first summation measures the weighted overall cost of matching the probe feature set with the gallery feature set based on the geometric and the textural information. The second summation penalizes the case where only few point correspondences are established, and the third summation makes point correspondence fuzzy, *i.e.*, $\mathbf{M}_{ij}$ can be any value between 0 and 1. Parameter C controls the fuzziness of correspondence matrix, *e.g.*, as the value of C gradually decreases, $\mathbf{M}_{ij}$ moves towards to either 0 or 1, such that the correspondence between two point sets becomes more definite. τ , λ_1 and λ_2

are parameters which control trade-offs between these penalties. The correspondence matrix M and transformation function f are then alternatively updated in an annealing process. The proposed MLERPM achieved promising results on partial face recognition. However, no constraint is enforced on the affine transformation matrix, which may generate unrealistic warping.

2.2.2 Point Set Matching and Graph Matching

Point set matching [107, 108, 109, 110, 111, 112, 113, 114] and graph matching [115, 116, 117, 118, 119, 120, 121, 122, 123, 124] are two related fundamental problems in computer vision and pattern recognition. Both of these two groups of approaches seek to establish correspondences between two matching feature sets.

The state-of-the-art point set matching techniques include [109, 110, 111, 112, 113, 114, 125]. Chui and Rangarajan [109] presented a feature set matching approach named TPS-RPM to align two point sets according to their geometric distribution by learning a non-affine transformation function embedded in a deterministic annealing process. Given two matching feature point sets, $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m]$ and $Y = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n]$, TPS-RPM learns a non-affine transformation function $f(\mathbf{x})$ and a correspondence matrix M alternatively to minimize the overall geometric distribution distance. The objective function is formulated as

$$E(M, f) = \sum_{i,j} m_{ij} \|\mathbf{y}_j - f(\mathbf{x}_i)\|^2 + \lambda \|Lf\|^2 + T \sum_{i,j} m_{ij} \log m_{ij} - \zeta \sum_{i,j} m_{ij} \quad (2.17)$$

where $m_{ij} = 1$ matches $\mathbf{x}_i$ to $\mathbf{y}_j$, and λ, T, ζ are the trade-off parameters. The first term minimizes the geometric distance; the second term constrains the energy of the TPS non-affine transformation; the third term calculates the negative entropy of M , *i.e.*, when T reaches zero, the fuzzy correspondence matrix M becomes binary. To derive the transformation function $f(\mathbf{x})$ and the correspondence matrix M , TPS-RPM

alternatively updates f and M while gradually decreasing the value of T .

Grauman and Darrell [110] presented a pyramid matching kernel which derives multi-resolution histograms from feature sets and measures feature set similarity by weighted histogram intersection. Jian and Vemuri [111] deployed Gaussian mixture models to represent the input point sets, and reformulated the problem of point set registration as the problem of aligning two Gaussian mixtures. Maier-Hein *et al.* [112] presented a convergent iterative closest point algorithm to accommodate anisotropic and inhomogenous localization error. Pokrass *et al.* [125] treated shape correspondence as a sparse modeling problem, and an outlier matrix is introduced to deal with the noisy points. The above-mentioned feature set matching approaches are not directly applicable for face recognition as most of them utilize either geometric or textural information of local features for matching. Alternatively, Li *et al.* [126] proposed a linear programming framework for feature matching, where both the geometric and textural features are employed. Their method is designed to match objects which are locally rigid and globally non-rigid, whereas human faces are globally rigid and locally non-rigid.

The idea of feature set matching has also been exploited in face recognition. For example, [127] used graph matching for face recognition. They first manually labeled face landmarks and then computed the face similarity based on local features around landmarks. In contrast, our partial face recognition approaches are fully automatic, which is free from manual labeling.

Graph matching is different from feature set matching in that the point set not only consists of feature points (nodes), but also the point-to-point connections (edges). Therefore, graph matching approaches consider both the geometric distribution between the matching point sets, and the pairwise edge consistency. Nonetheless, these edge constraints increase the computational load substantially. Therefore, our approach RPSM doesn't consider the edge information.

2.2.3 Discussion

Some of the aforementioned face recognition works can be applied to the partial face recognition problem. For instance, the existing SRC based methods have achieved excellent results in recognizing occluded faces [18, 19, 20, 27]. These works may fail if the probe faces are arbitrary partial faces. This is because the prerequisite face alignment process becomes largely unreliable. To address this problem, two partial face recognition approaches [5, 106] have been proposed, but there still exists room for improvement.

Our RPSM approach deals with the facial occlusion and arbitrary face patch jointly by aligning the partial face to the gallery face first and subsequently identifying it. In particular, RPSM converts the partial face alignment problem as a local feature set matching problem. Compared to other feature set matching approaches, RPSM utilizes both the textural and geometric information during the matching, thereby achieving more robust and accurate alignment.

2.3 Conclusions

Both face alignment and robust face recognition fields have made tremendous progress during the past decades. For instance, the state-of-the-art landmark detection approach is capable of detecting landmarks at 3000 FPS [37] and several face recognition methods [24, 25] have outmatched humans on the challenging LFW dataset [1]. Nevertheless, there still exists some room for improvement.

For the holistic face alignment, the performance of existing facial landmark detection approaches are still unsatisfactory in dealing with large pose variations and drifted bounding boxes. Specifically, more robust pose initialization schemes are desirable to deal with the vast range of in-plane and out-of-plane rotation, as well as scale variations and face translation. Moreover, alignment accuracy can be further improved by learn-

ing more expressive pose-informative features and more robust feature-to-pose mapping function. To this end, we have designed two deep learning based landmark detection approaches in Chapter 3 and Chapter 4, respectively, which achieve superior performance on three benchmarks.

In terms of the partial face alignment, several existing holistic face alignment methods can be applied on the occluded partial faces, provided that the area of occlusion is moderate and the faces are near upright. For arbitrary partial faces, their performance will degrade greatly (this claim is verified by our experiments in Chapter 5). For partial face recognition, existing SRC based methods have achieved excellent results on identifying occluded faces [18, 19, 20, 27]. However, these works may not work well on arbitrary partial faces. To address this partial face alignment and recognition problem, we devise an approach based on robust feature set matching, where both alignment and recognition are achieved jointly in the same framework. Extensive experiments on both partial face alignment and recognition validate the efficacy of our approach in Chapter 5.

Chapter 3

Holistic Face Alignment Using CDAN

Face alignment (*a.k.a.* facial landmark detection) has been a popular research topic over the past few years due to its importance in various face analysis applications such as face recognition, facial expression classification, and age estimation. Current holistic face alignment approaches have achieved promising results, but there still exists room for improvement, especially when facial images are captured under uncontrolled conditions. In this chapter, we propose a new Cascaded Deep Auto-encoder Networks (CDAN) approach for holistic face alignment, where a Global Exemplar-based Deep Auto-encoder Network (GEDAN) is designed to better initialize the face pose, and a Localized Deep Auto-encoder Network (LDAN) is proposed to accurately update the landmark positions.

3.1 Introduction

The face alignment problem is inherently non-convex, and most approaches tackle it using two stages, namely a pose initialization stage and an iterative pose update stage. Zhang *et al.* [34] devised a Coarse-to-Fine Auto-encoder Networks (CFAN) for face alignment. For pose initialization, CFAN takes the holistic facial image as input and generates a coarse pose prediction by a global Stacked Auto-encoder Network (SAN).

Figure 3.1: The pipeline of our Cascaded Deep Auto-encoder Networks (CDAN) for facial landmark detection. The facial landmarks positions are gradually refined as the image passes through GEDAN and several LDANs from the left to the right. Meanwhile facial image resolution is increased as well, such that more details are incorporated for face alignment. Specifically, we have $S_{i+1} = S_i + \Delta S_{i+1}$, where S_0 is the shape prediction of GEDAN.

For pose update, CFAN utilizes pose-indexed SIFT features which are extracted from pose-indexed patches as feature representations of landmarks, and deploys Local SAN to model the relationship between local features and facial landmark poses. Their model fails to extract pose-informative local features because all SIFT features are simply concatenated together to learn the deep model.

Inspired by CFAN [34], we also deploy a Deep Auto-encoder Network (DAN) to initialize the facial pose. However, regressing on holistic images poses a great difficulty for deep model. Specifically, conventional DAN is incapable of estimating near-profile pose accurately. To better initialize facial poses, we propose a Global Exemplar-based Deep Auto-encoder Network (GEDAN) which takes the holistic facial image as input and generates a roughly aligned face configuration for the following pose update stage. GEDAN incorporates several exemplars at the top layer to form a non-linear regression model, which enlarges the deep auto-encoder network’s capacity in pose estimation. Compared to DAN [34], our GEDAN can better handle large pose variations.

In terms of the pose update stage, we further devise a Localized Deep Auto-encoder Network (LDAN) for face alignment. Specifically, the first layer of our LDAN model

consists of individual Local Auto-Encoders (LAEs), each of which aims to extract localized pose-informative features from the corresponding pose-indexed patches. The outputs of these LAEs are then directly fed to their corresponding local regressors. Furthermore, we concatenate the outputs of these LAEs into a global feature vector and encode them by using several layers of auto-encoders. Subsequently, the global features are led to local regressors to impose global facial structure constraints on local regressors. Hence, pose-informative features are extracted in a task-driven manner, and they are directly led to landmark detection under a global facial structure constraint. Assembling them together, we cascade GEDAN with a series of LDANs to form Cascaded Deep Auto-encoder Networks (CDAN) for accurate face alignment. Fig. 3.1 illustrates the basic idea of our CDAN. Experimental results on three datasets show that our CDAN achieves superior face alignment accuracy with real-time speed.

The main contributions of this approach are summarized as follows:

- We propose a new deep auto-encoder network called GEDAN to provide a promising starting facial pose for face alignment. GEDAN incorporates several exemplars at the regression layer, and it's able to effectively estimate near-profile poses or facial poses under extreme illumination.
- We devise a novel deep model LDAN to better extract pose-informative features from local patches, and to efficiently model the highly nonlinear relationship between local features and face poses.

3.2 Cascaded Deep Auto-encoder Network

In this section, we describe the details of our CDAN model. We first present the overall pipeline of CDAN, and then detail the structures as well as the learning processes of GEDAN and LDAN. Lastly, we discuss the implementation details of the CDAN approach.

Fig. 3.1 shows the basic idea of our approach. First, the detected low-resolution facial image is fed to GEDAN to obtain a preliminary facial landmark configuration S_0 . S_0 is then taken as the initial shape for the first LDAN which refines the shape to $S_1 = S_0 + \Delta S_1$. Similarly, the following LDANs take the regression result of the preceding LDAN as their initial shape and perform local adjustments. Meanwhile, as the pipeline proceeds, image resolution is gradually increased. Therefore, more details are harnessed to further increase the detection accuracy. The difference between our CDAN and CFAN [34] lies on the specific design of component networks, which will be detailed shortly.

3.2.1 Global Exemplar-based Deep Auto-encoder Network

Most of the previous approaches initialize the facial pose either as mean shape (*e.g.*, AAM models [9, 28], SDM [8]) or with multiple random shapes [6, 33] fit in the detected bounding box. These initialization schemes are prone to lead the alignment process to local minimum when dealing with near-profile pose or extreme illumination. Differently, CFAN [34] utilizes a global SAN to obtain an approximate pose prediction based on a low-resolution image, thereby achieving better pose initialization. Nonetheless, directly inferring pose from holistic image is quite challenging. Given limited training samples, the global SAN's performance is unsatisfactory in estimating near-profile pose.

To improve the pose estimation accuracy on holistic facial images, we propose a Global Exemplar-based Deep Auto-encoder Network (GEDAN). GEDAN samples several exemplars from a joint pose appearance space and deploys them to assist pose estimation. Fig. 3.2 shows the structure of our GEDAN. The hallmark of the GEDAN is its top regression layer which consists of two modules, namely a linear regression module and a nonlinear exemplar-based regression module. The linear module aims to estimate the facial pose from a global perspective while the exemplar-based nonlinear module strives at adjusting the regression result from a local perspective. With the help from

Figure 3.2: The architecture of our Global Exemplar-based Deep Auto-encoder Network (GEDAN). The first three layers are auto-encoder layers, which progressively encode the raw facial image vector $H^{(0)}$ to pose-correlated features $H^{(3)}$. $H^{(3)}$ is then fed to pose regressor to obtain a preliminary pose result S_0 . This pose regressor is composed of a global linear regressor W^G and K exemplar based nonlinear regressor.

these chosen exemplars, facial poses can be inferred more accurately from the holistic image.

Assume we have a training set Ω consisting of various training samples x_i with the corresponding facial shape $S_i = [p_1; p_2; \dots; p_M]$, where M is the number of facial landmarks. p_l is the geometric position of the l th landmark, and $S \in \mathbb{R}^{2M \times 1}$. For each x_i , it has a corresponding raw pixel vector $H_i^{(0)}$.

The overall objective function for GEDAN is formulated as follows:

$$J = \min_{f,r} \frac{1}{2} \sum_{\mathbf{x}_i \in \Omega} \|\mathbf{S}_i - r(f(\mathbf{H}_i^{(0)}))\|_2^2 + \frac{\lambda}{2} \|\Theta(f)\|_2^2, \quad (3.1)$$

where function $f(\mathbf{H}_i^{(0)})$ is the overall embedding function of the first three layers. Specifically, $f(\mathbf{H}_i^{(0)})$ encodes the raw pixel vector $\mathbf{H}_i^{(0)}$ to pose-correlated features $\mathbf{H}_i^{(3)}$. Function $r(\mathbf{H}_i^{(3)})$ then infers pose $\mathbf{S}_i$ from this embedded features. $\Theta(f)$ is the coding parameters of deep auto-encoder network. The first term of Eq. (4.1) endeavors to minimize the training error, and the second term of Eq. (4.1) regularizes the learning process to control the scales of GEDAN's parameters.

For the j th coding layer, $1 \leq j \leq 3$, its output $\mathbf{H}^{(j)}$ is

$$\mathbf{H}^{(j)} = s(\mathbf{W}^{(j)}\mathbf{H}^{(j-1)} + \mathbf{b}^{(j)}), \quad (3.2)$$

where $s(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{x}}}$ is an element-wise sigmoid function. Note that pixel intensity values of all training images need to be normalized to $[0, 1]$, such that they are compatible with auto-encoder.

Now, let's focus on function $r(\mathbf{h}_i)$ at the regression layer (here we use $\mathbf{h}_i$ instead of $\mathbf{H}_i^{(3)}$ for notation clarity). Most of the previous approaches use simple regression methods for $r(\mathbf{h}_i)$, *e.g.*, linear regression models, which are insufficient in modeling the complex feature-to-pose relationship. To better model this relationship, we propose an exemplar-based regression model. Its regression function is described as follows

$$r(\mathbf{h}_i) = \mathbf{W}^G \mathbf{h}_i + \mathbf{b}^G + \mathbf{Q}\Phi(\mathbf{h}_i), \quad (3.3)$$

where $\mathbf{W}^G$ and $\mathbf{b}^G$ are the parameters of the linear regression model, and $\mathbf{Q}\Phi(\mathbf{h}_i)$ is the nonlinear exemplar-based module. Specifically, $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_K]$ is a $2M \times K$ matrix, and $\Phi(\mathbf{h}_i)$ is a $K \times 1$ vector which describes the local geometric structure of the

Figure 3.3: Some sampled exemplars, each of which represents a distinct pose-appearance pair. Note that the actual exemplar's appearance features are the embedded features at the third layer, here we show their corresponding facial images.

embedded face manifold at the position of $\mathbf{h}_i$:

$$\Phi(\mathbf{h}_i) = [e^{\left(-\frac{(\mathbf{h}_i - \mathbf{a}_1)^2}{2\sigma^2}\right)}; e^{\left(-\frac{(\mathbf{h}_i - \mathbf{a}_2)^2}{2\sigma^2}\right)}; \dots; e^{\left(-\frac{(\mathbf{h}_i - \mathbf{a}_K)^2}{2\sigma^2}\right)}], \quad (3.4)$$

where $\mathbf{a}_k$ is the embedded features of the k th exemplar at the third layer.

We can view this nonlinear model $\mathbf{Q}\Phi(\mathbf{h}_i)$ as the overall guidance offered by K trained exemplar experts for pose estimation. Then $\mathbf{q}_k$ is the advice given by the k th exemplar expert, and $e^{\left(-\frac{(\mathbf{h}_i - \mathbf{a}_k)^2}{2\sigma^2}\right)}$ calculates the confidence of the k th exemplar expert in dispensing advice, *e.g.*, the expert will be confident if the testing image has large resemblance with him/her in terms of appearance.

Algorithm 1: GEDAN training

```

Initialize  $\mathbf{W}^G$ ,  $\mathbf{b}^G$  and  $\mathbf{Q}$  to be all zero matrix;
Initialize  $\mathbf{W}^{(j)}$  and  $\mathbf{b}^{(j)}$ ,  $1 \leq j \leq 3$  using pre-training;
for  $Epoch = 1 : T$  do
    Fix  $\mathbf{Q}$ ;
    Update  $\mathbf{W}^G$ ,  $\mathbf{b}^G$ ,  $\mathbf{W}^{(j)}$  and  $\mathbf{b}^{(j)}$  using BP;
end
Conduct K-means clustering with Eq. (3.5), get the  $K$  exemplars.
for  $Epoch = T + 1 : N$  do
    Calculate  $\Phi(\mathbf{h}_i)$  with Eq. (3.4) for all training data;
    Update  $\mathbf{Q}$ ,  $\mathbf{W}^G$ ,  $\mathbf{b}^G$ ,  $\mathbf{W}^{(j)}$  and  $\mathbf{b}^{(j)}$  using BP;
end

```

The K chosen exemplars represent the overall distribution of training data, *e.g.*, they can be the centroids of various clusters, each of which represents a unique setting of facial pose and illumination. To select exemplars from training samples, we simply use the K-means clustering algorithm. Instead of singly clustering on embedded features $\mathbf{h}$ or on facial poses, we propose to jointly cluster on both $\mathbf{h}$ and the corresponding face poses $\mathbf{S}$:

$$\underset{\Omega}{\operatorname{argmin}} \sum_{k=1}^K \sum_{\{\mathbf{x}, \mathbf{S}\} \in \Omega_k} (\|f(\mathbf{x}) - \mathbf{a}_k\|^2 + \tau \|\mathbf{S} - \bar{\mathbf{S}}_k\|^2) \quad (3.5)$$

where $\Omega = \{\Omega_1, \Omega_2, \dots, \Omega_K\}$ is a partition of all training samples into K sets; $\mathbf{a}_k$ and $\bar{\mathbf{S}}_k$ are the pose-correlated features and mean shape of the centroid of set Ω_k respectively. τ is simply set as the ratio between the variance of $f(\mathbf{x})$ and the variance of $\mathbf{S}$. The value of K is set by conducting validation experiments which will be detailed in the Experiment session. Note that we choose to cluster on embedded features $f(\mathbf{x})$ instead of on raw pixels $\mathbf{x}$. This is because pose-correlated features $f(\mathbf{x})$ have higher correlation with pose than noisy raw pixel features do. Moreover, facial poses are incorporated into the clustering process as well. It's desirable for the exemplars to span a wide range of pose variations, such that the exemplars may have more "experience" in dealing with large head poses. Fig. 3.3 shows some sampled exemplars. These exemplars span a large

variation of pose-appearance space. Note that using exemplars is not a new idea, *e.g.*, Liao *et al.* [128] applied exemplars to assist end-to-end face recognition. The novelty of our work is we introduce exemplars to auto-encoder network, thereby enhancing the network's capacity.

To train this GEDAN model, we first initialize parameters of the first three layers $\mathbf{W}^{(j)}$ and $\mathbf{b}^{(j)}$ by unsupervised pre-training, and set regression parameters $\mathbf{W}^G, \mathbf{b}^G, \mathbf{Q}$ to be all zero matrix. Then we fix $\mathbf{Q}$ and update the rest parameters using standard back-propagation (BP) [129]: we feed-forward the input image throughout the deep network and update parameters with the stochastic gradient descent algorithm. Next, we include $\mathbf{Q}$ in the update process. To begin with, we perform K-means clustering on the joint pose-appearance space using Eq. (3.5). Then we use the BP algorithm to update all these parameters together. The detailed training process of GEDAN is shown in Algorithm 1. Note that the K-means clustering is only conducted once during the training. This is because after the first T epochs of training, the learned features $\mathbf{h}_i$ are informative enough to deduce representative exemplars. Moreover, conducting K-means after each epoch might result in different sets of exemplars, which brings oscillation to the training process.

3.2.1.1 The calculations of GEDAN gradients

The details of GEDAN back propagations (BP) are described in this section. Here we focus on the BP process where $\mathbf{Q}$ is included for joint update after exemplar selection. The BP process updates the parameters layer by layer from the top to the bottom. Hence, we deduce the partial derivative of J with respect to $\mathbf{W}_G, \mathbf{Q}, \mathbf{h}_i$ and α_k first. According to the chain rule of partial derivatives, the partial derivatives of $\frac{\partial J}{\partial \mathbf{W}_G}$ and $\frac{\partial J}{\partial \mathbf{Q}}$ are given

by

$$\frac{\partial J}{\partial \mathbf{W}_G} = \sum_i (\mathbf{r}_i - \mathbf{S}_i) \mathbf{h}_i^T, \quad (3.6)$$

$$\frac{\partial J}{\partial \mathbf{Q}} = \sum_i (\mathbf{r}_i - \mathbf{S}_i) \Phi(\mathbf{h}_i)^T, \quad (3.7)$$

where $\mathbf{r}_i$ represents $r(\mathbf{h}_i)$. Similarly, the partial derivative $\frac{\partial J}{\partial \mathbf{h}_i}$ is computed as follows:

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{h}_i} &= \frac{\partial J}{\partial \mathbf{r}_i} \frac{\partial \mathbf{r}_i}{\partial \mathbf{h}_i} = (\mathbf{r}_i - \mathbf{S}_i)^T \frac{\partial \mathbf{r}_i}{\partial \mathbf{h}_i}, \\ \frac{\partial \mathbf{r}_i}{\partial \mathbf{h}_i} &= \mathbf{W}^G + \mathbf{Q} \frac{\partial \Phi}{\partial \mathbf{h}_i}, \\ \frac{\partial \Phi}{\partial \mathbf{h}_i} &= -\Phi \mathbf{e}^T \odot \left[\frac{(\mathbf{h}_i - \mathbf{a}_1)^T}{\sigma^2}; \dots; \frac{(\mathbf{h}_i - \mathbf{a}_K)^T}{\sigma^2} \right]. \end{aligned} \quad (3.8)$$

where $\mathbf{e}$ is an all one vector having the same dimension as $\mathbf{h}_i$'s, and $\odot$ is an element-wise multiplication operator.

In terms of $\frac{\partial J}{\partial \alpha_k}$, since it participates in the computation of each $\Phi(\mathbf{h}_i)$, it is formulated as

$$\begin{aligned} \frac{\partial J}{\partial \alpha_k} &= \sum_i \frac{\partial J}{\partial \mathbf{r}_i} \frac{\partial \mathbf{r}_i}{\partial \alpha_k} = \sum_i (\mathbf{r}_i - \mathbf{S}_i)^T \frac{\partial \mathbf{r}_i}{\partial \alpha_k}, \\ \frac{\partial \mathbf{r}_i}{\partial \alpha_k} &= \mathbf{Q} \frac{\partial \Phi}{\partial \alpha_k}, \end{aligned} \quad (3.9)$$

where $\frac{\partial \Phi}{\partial \alpha_k}$ is a matrix whose k th row is $-\exp(-\frac{(\alpha_k - \mathbf{h}_i)^2}{2\sigma^2}) \frac{(\alpha_k - \mathbf{h}_i)^T}{\sigma^2}$, and all the rest elements equal to 0. By denoting $-\exp(-\frac{(\alpha_k - \mathbf{h}_i)^2}{2\sigma^2}) \frac{(\alpha_k - \mathbf{h}_i)}{\sigma^2}$ as $\mathbf{u}_{ki}$ and the k th column of $\mathbf{Q}$ as $\mathbf{q}_k$, we have

$$\begin{aligned} \frac{\partial \mathbf{r}_i}{\partial \alpha_k} &= \mathbf{q}_k \mathbf{u}_{ki}^T, \\ \frac{\partial J}{\partial \alpha_k} &= \sum_i \frac{\partial J}{\partial \mathbf{r}_i} \frac{\partial \mathbf{r}_i}{\partial \alpha_k} = \sum_i (\mathbf{r}_i - \mathbf{S}_i)^T \mathbf{q}_k \mathbf{u}_{ki}^T. \end{aligned} \quad (3.10)$$

For the rest derivatives, we only present the computation of $\frac{\partial J}{\partial \mathbf{W}^{(3)}}$. The others can be deduced in a similar manner. In particular, $\frac{\partial J}{\partial \mathbf{W}^{(3)}}$ is given by

$$\frac{\partial J}{\partial \mathbf{W}^{(3)}} = \sum_i \frac{\partial J}{\partial \mathbf{h}_i} \frac{\partial \mathbf{h}_i}{\partial \mathbf{W}^{(3)}} + \sum_k \frac{\partial J}{\partial \alpha_k} \frac{\partial \alpha_k}{\partial \mathbf{W}^{(3)}}, \quad (3.11)$$

where both $\frac{\partial \mathbf{h}_i}{\partial \mathbf{W}^{(3)}}$ and $\frac{\partial \alpha_k}{\partial \mathbf{W}^{(3)}}$ can be derived with ease. Thus, they're not specified here.

3.2.2 Localized Deep Auto-encoder Network

While GEDAN is able to estimate facial pose from a face image, its accuracy is still far from satisfactory. To further improve the alignment accuracy, we devise a new deep auto-encoder structure called Localized Deep Auto-encoder Network (LDAN). Fig. 3.4 shows the structure of our LDAN. In the proposed LDAN model, there are four layers. The bottom three layers are encoding layers, and the top one is the regression layer. Specifically, a series of Local Auto-Encoders (LAEs) at the first layer are employed to extract pose-informative features from local patches. The outputs of these LAEs are then concatenated and further encoded at Layer 2 and Layer 3 to form a global shared feature. In the regression layer, both local features from Layer 1 and global structural features from Layer 3 are combined for regression.

We use pose-indexed features [130] as the input for our LDAN model. Pose-indexed feature has been widely applied in many previous face alignment methods [6, 8, 34, 37]. The existing methods use random ferns [6, 36] and concatenated SIFT features [8, 34] as pose-indexed features. Random fern extracts local features from sparse locations surrounding facial landmarks, which has limited capacity in describing local patches. Concatenated SIFT features simply joins pose-indexed SIFT features into a single feature vector. While concatenation of local features implicitly encodes global facial structure, it might lose some crucial local information. To achieve more accurate face alignment, pose-sensitive local features need to be extracted.

Figure 3.4: The architecture of a Localized Deep Auto-encoder Network (LDAN). In the first layer, the i th pose-indexed patch (pose-indexed patch is a local patch extracted from the corresponding facial landmark) is encoded by a Local Auto-Encoder (LAE) $W_i^{(1)}$, and the output is $h_i^{(1)}$. The output of LAEs proceed in two paths. On one hand, each $h_i^{(1)}$ is directly fed into a local regressor R_i to predict the corresponding landmark's position deviation Δp_i . On the other hand, these outputs are concatenated into a single global feature vector $H^{(1)}$ which is taken as the input for the second layer. Therefore, the global facial structure can be implicitly preserved. Finally, the output $H^{(3)}$ of the top layer is linked to all local regressors, where both global facial structural information and local landmark feature information are jointly exploited for facial landmark detection.

After feature extraction, local features need to be integrated to encode the global facial structure, thereby refraining from arbitrary pose regression. This is because some landmarks' positions are ambiguous from a local patch perspective. This is illustrated

Figure 3.5: For the right face image: the blue point is the current estimated position of a cheek landmark, and the green point is its ground-truth position. The patch in the left-hand side is its enlarged view, which shows the insufficiency of a single patch on dealing with ambiguous landmark. Intuitively, based on this patch alone, the local regressor might adjust the current estimated position to its nearest edge point (the direction is marked in red arrow to the red point). However, its ground-truth position (green dot) is far away from the estimated position (red point). Thus when the position of a facial landmark is uncertain from a local patch perspective, landmark detection using that local patch alone cannot work well. (Best view in the color pdf file.)

in Fig. 3.5. Moreover, if a facial landmark is occluded, local features extracted from the occluded patch are unreliable for detection of this landmark.

Motivated by the above observations, we propose a Localized Deep Auto-encoder Network (LDAN) for face alignment. (1) To extract pose-informative features from local patches, a number of Local Auto-Encoders (LAEs) are configured to separately learn position-specific features. For example, the LAE for the eye corner and that for the nose tip are trained separately to learn different features to reflect semantic information of their landmarks. By feeding the output of LAEs to the corresponding regressors, local pose-informative features are directly led to adjust landmark position. Conversely, the training pose deviations are back-propagated to adjust the parameters of LAE, driving LAE to extract more pose-correlated local features. Fig. 3.6 visualizes the weight corresponding to four facial landmarks. Clearly, LAEs extract pose-informative features from local patches by learning the inherent local facial structure. (2) To preserve facial structural information, the outputs of LAEs from all local patches are concatenated

Figure 3.6: From top to the bottom: the weight visualization of the 49th [3] LAE which corresponds to the left outer corner of the mouth; the 4th LAE which corresponds to the left cheek; the 40th LAE which is the left eye corner; and the 9th LAE which is the chin landmark.

into a single feature vector. This feature vector is further encoded by various layers of auto-encoders. By doing so, local features can be gradually integrated into global shared features which are then fed into all local regressors. Therefore, local regressor R_i not only utilizes pose-informative features extracted from its corresponding patch, but also integrates global information in the regression procedure. Thus, more accurate pose adjustment can be achieved.

Let $\bar{S}$ be the initialized shape (the pose estimation result from GEDAN or from the proceeding LDAN), then the ground-truth pose difference is $\Delta S = S - \bar{S}$. The overall

objective function for LDAN is

$$J = \min_{\Phi} \frac{1}{2} \sum_{\mathbf{x} \in \Omega} \|\Delta \mathbf{S} - \Phi(\mathbf{x})\|_2^2 + \frac{\lambda}{2} \|\Theta(\Phi)\|_2^2, \quad (3.12)$$

where the system function $\Phi(\mathbf{x})$ maps the local features to pose deviation $\Delta \mathbf{S}$ and $\Theta(\Phi)$ is the parameter of LDAN. The first term of Eq. (3.12) minimizes the training error, and the second term of Eq. (3.12) regularizes the learning process to control the scales of LDAN's parameters. To solve the optimization problem, we use the back-propagation method [129]. For notation clarity, we take a single training sample $\mathbf{x}$ as an example.

Layer 1: Let $\mathbf{f}_i$ be the pose-indexed local features (*e.g.*, SIFT or raw pixel vector) from the local patch extracted from the i th landmark, and $\{\mathbf{W}_i^{(1)}, \mathbf{b}_i^{(1)}\}$ be the corresponding parameter of LAE, $1 \leq i \leq M$. The output of the i th LAE at Layer 1 is

$$\mathbf{h}_i^{(1)} = s \left(\mathbf{W}_i^{(1)} \mathbf{f}_i + \mathbf{b}_i^{(1)} \right). \quad (3.13)$$

Layer 2: The output of all LAEs $\mathbf{h}_i^{(1)}$ at Layer 1 are concatenated into a global feature vector as the input for the auto-encoder at Layer 2:

$$\begin{aligned} \mathbf{H}^{(1)} &= [\mathbf{h}_1^{(1)}; \mathbf{h}_2^{(1)}; \dots; \mathbf{h}_M^{(1)}], \\ \mathbf{H}^{(2)} &= s \left(\mathbf{W}^{(2)} \mathbf{H}^{(1)} + \mathbf{b}^{(2)} \right). \end{aligned} \quad (3.14)$$

Meanwhile, $\mathbf{h}_i^{(1)}$ is connected to local regressor $\mathbf{R}_i$ to adjust the i th landmark position.

Layer 3: $\mathbf{H}^{(2)}$ of the second layer is used as the input for the auto-encoder of Layer 3:

$$\mathbf{H}^{(3)} = s \left(\mathbf{W}^{(3)} \mathbf{H}^{(2)} + \mathbf{b}^{(3)} \right). \quad (3.15)$$

Layer 4: Layer 4 is the regression layer which connects both the global shared fea-

tures $\mathbf{H}^{(3)}$ and local pose-informative features $\mathbf{h}_i^{(1)}$ to the local regressor $\mathbf{R}_i$. Therefore, the output of the regressor $\mathbf{R}_i$ is computed as

$$\Delta \tilde{\mathbf{p}}_i = \mathbf{W}_i^L \mathbf{h}_i^{(1)} + \mathbf{b}_i^L + \mathbf{W}_i^G \mathbf{H}^{(3)} + \mathbf{b}_i^G. \quad (3.16)$$

The overall landmark variation prediction is

$$\Delta \tilde{\mathbf{S}} = [\Delta \tilde{\mathbf{p}}_1; \dots; \Delta \tilde{\mathbf{p}}_i; \dots; \Delta \tilde{\mathbf{p}}_M]. \quad (3.17)$$

We use the back-propagation algorithm to update parameters $\{\mathbf{W}_i^G, \mathbf{b}_i^G, \mathbf{W}_i^L, \mathbf{b}_i^L\}$ and $\{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}$, where $2 \leq l \leq 3$. Here we focus on the update of the parameter of the i th LAE, *i.e.*, $\mathbf{W}_i^{(1)}$. There are two paths of gradient information passing down from the top regression layer to the i th LAE. One path is through $\mathbf{W}_i^G$, $\mathbf{W}^{(3)}$ and $\mathbf{W}^{(2)}$, gradient information passing along this path is denoted as $\frac{\partial J^G}{\partial \mathbf{h}_i^{(1)}}$. The other path is through $\mathbf{W}_i^L$, which is defined as $\frac{\partial J^L}{\partial \mathbf{h}_i^{(1)}}$.

Thus, the gradient of the objective function over the parameter $\mathbf{W}_i^{(1)}$ is given by

$$\frac{\partial J}{\partial \mathbf{W}_i^{(1)}} = \frac{\partial J^L}{\partial \mathbf{h}_i^{(1)}} \frac{\partial \mathbf{h}_i^{(1)}}{\partial \mathbf{W}_i^{(1)}} + \frac{\partial J^G}{\partial \mathbf{h}_i^{(1)}} \frac{\partial \mathbf{h}_i^{(1)}}{\partial \mathbf{W}_i^{(1)}} + \lambda \mathbf{W}_i^{(1)}, \quad (3.18)$$

where $\lambda \mathbf{W}_i^{(1)}$ is the gradient of weight decaying term. Since $\frac{\partial J^L}{\partial \mathbf{h}_i^{(1)}} = (\mathbf{W}_i^L)^T (\Delta \tilde{\mathbf{p}}_i - \Delta \mathbf{p}_i)$,

$$\begin{aligned} \frac{\partial J^L}{\partial \mathbf{W}_i^{(1)}} &= \frac{\partial J^L}{\partial \mathbf{h}_i^{(1)}} \frac{\partial \mathbf{h}_i^{(1)}}{\partial \mathbf{W}_i^{(1)}} = \\ &= \left((\mathbf{W}_i^L)^T (\Delta \tilde{\mathbf{p}}_i - \Delta \mathbf{p}_i) \odot (\mathbf{e} - \mathbf{h}_i^{(1)}) \odot (\mathbf{h}_i^{(1)}) \right) \mathbf{f}_i^T, \end{aligned} \quad (3.19)$$

where $\odot$ is element-wise multiplication operator. $\mathbf{e}$ is a vector with the same size as $\mathbf{h}_i^{(1)}$ and each element of $\mathbf{e}$ is set as 1. $\frac{\partial J^G}{\partial \mathbf{W}_i^{(1)}}$ can be computed in the same way.

Approach	CDAN	CFAN
Global Model	Applies exemplar-based non-linear regression at the top layer	Plain auto-encoder network
Local Model	Uses both pose-indexed SIFT features and raw pixels features; Employs LAEs to project the features to low dimension; Direct connection from the output of LAEs to the top layer	Uses pose-indexed SIFT features; Utilizes PCA for dimension reduction.

Table 3.1: The difference between CDAN and CFAN.

3.2.3 Difference with CFAN

CFAN [34] uses Stacked Auto-encoder Network (SAN) for face alignment. In particular, it consists of the Global-SAN which initializes the facial pose, and a series of Local-SANs which gradually update the pose. The Global-SAN is similar with the GEDAN except that it doesn't employ exemplars to assist pose estimation. However, directly regressing on holistic faces is highly challenging, and the Global-SAN is incompetent to infer near-profile facial pose. This problem is further exacerbated by the scarcity of large-pose faces in the training data. Differently, GEDAN adopts multiple exemplars to assist alignment. These exemplars represent a vast range of illumination conditions and facial poses, and the incorporation of them enlarges the capacity of the auto-encoder network.

For pose update, CFAN extracts local features (SIFT) from pose-indexed patches, and concatenates all these local features into a single feature vector. This concatenated feature is further projected to a low dimensional space by PCA, which is then propouted through Local-SAN for regression. While CFAN has put some emphasis on local features, all these local features are mixed together by various layers (PCA embedding, auto-encoder nonlinear projection) before regression. Thus the top regression layer of

SAN is dealing with global mixed features, where some pose-informative features might be lost after passing through these layers. On the contrary, our LDAN extracts pose-correlated features in a data-driven way by LAE, and local features are connected to local regressor directly before being mixed into global shared features. Hence, crucial local details are kept as much as possible for regression. Moreover, the problem of vanishing gradients of deep neural network [131] during back-propagation is mitigated as the local regressor directly feeds back the gradient information to LAE, speeding up the training process. Table 3.1 lists the difference between these two approaches.

3.3 Implementation Details

We use 68 landmarks [132] to annotate each face image, and Fig. 3.14 shows some example images of the annotated landmarks. Our CDAN consists of one GEDAN followed by three LDANs. For the GEDAN, we sampled 50 exemplars ($K = 50$) from the pose-appearance space and we set σ to 10^1 . In terms of LDANs, they shared the same network configuration. Specifically, there were 20 hidden units for each LAE in the first layer, 1000 and 500 hidden units for the second and the third layer respectively. For both GEDAN and LDANs, the parameters of auto-encoders were pre-trained by using RBM [133]. Specifically, we trained the model in a layer-wise manner. The learning rate was set to 0.01 and momentum was set to 0.9. The training data for the pre-training phase is as same as the training data we adopted, which will be detailed in the following section. In terms of local features, we employed pose-indexed SIFT features as the input for the first two LDANs, and raw pixel vectors of pose-indexed patches for the third LDAN. This is because for the first two LDANs, the search range of each landmark was very large, where SIFT features were more compact than raw pixel vectors. While for the third LDAN, the search range was small enough so that raw pixel vector was as

¹The values of K and σ were set according to a validation experiment.

Figure 3.7: The overall NRMSE distributions of GEDAN, GSDAN and Global-SAN across LFPW, HELEN and IBUG testing datasets.

compact as SIFT descriptor. Moreover, it is more efficient to extract raw pixel vector than SIFT features. Therefore, we deployed pose-indexed raw pixel vectors as the local features for the third LDAN. The value of weight decay parameter λ for both GEDAN and LDAN was empirically set to 0.0001.

3.4 Experiments

In this section, we first describe the datasets and experimental settings. Then we investigate the performance of a single GEDAN and a single LDAN separately. Subsequently, we examine the pose estimation accuracy versus different number of LDANs. Lastly, we compare our CDAN with several state-of-the-art face alignment approaches.

(a) LFPW

(b) HELEN

(c) IBUG

Figure 3.8: CED curve of a single GEDAN on the LFPW, HELEN and IBUG datasets, respectively.

3.4.1 Datasets and Experimental Settings

- LFPW [39]: There are 1432 facial images in the original LFPW dataset, which are acquired from the Internet. Each image is labeled with 35 facial landmarks. In the IBUG evaluation, these images are re-labeled with 68 landmarks. Since some URLs of the original LFPW images are invalid, there are only 811 training images and 224 testing images available on the IBUG website [3].
- HELEN [134]: There are 2000 training images and 330 testing images in the HELEN dataset, which are collected under wild (uncontrolled) conditions. Similar with LFPW, we employed the 68-landmark format for evaluation in our experiments, where the annotations are also provided by the IBUG website.
- AFW [7]: AFW dataset consists of 337 face images. These images are collected from the Internet under uncontrolled conditions. Similarly, we used the landmark annotations from the IBUG website, where each image has 68 landmarks.
- IBUG [132]: Compared to the LFPW and HELEN datasets, the IBUG dataset has larger variations on poses, illumination, and expressions of human faces. There are in total 135 images in this dataset.

For all these datasets, we used bounding boxes downloaded from IBUG website. These bounding boxes tightly enclose facial landmarks. To robustly estimate facial landmarks situated near the border, their corresponding neighborhood patches need to be incorporated in the bounding box. Hence, we enlarged the downloaded bounding boxes by 30% on both the width and height. We trained our CDAN model with the LFPW training images, HELEN training images, and the AFW dataset. In addition, we augmented the training images by flipping, random rotation and random crop. Having augmented the dataset, we had 18888 training samples to train our CDAN model. The Normalized Root Mean Squared Error (NRMSE) was applied to measure the error

K	0	10	20	50	80	120
NRMSE(%)	9.24	8.55	8.02	7.73	7.68	7.65

Table 3.2: The overall NRMSE of the GEDAN with various number of exemplars (K).

between estimated landmark positions and the ground truth. Specifically, NRMSE is defined as the average alignment error normalized by the Euclidean distance between the outer corners of the eyes [3]. Finally, the Cumulative Error Distribution (CED) curves of NRMSE were used to quantitatively evaluate the performance of different face alignment methods.

3.4.2 Evaluation of GEDAN

The hallmark of GEDAN is the incorporation of exemplar based non-linear regression module at the top layer, which enlarges the capacity of a deep auto-encoder network. In a similar spirit, we devised a baseline whose regression layer is

$$r(\mathbf{h}_i) = \mathbf{W}^G \mathbf{h}_i + \mathbf{b}^G + \mathbf{P}(\mathbf{h}_i \odot \mathbf{h}_i), \quad (3.20)$$

This baseline exploits the element-wise square of $\mathbf{h}_i$. Hence its regression layer is non-linear. We simply termed this baseline as “Global Square Deep Auto-encoder Network” (GSDAN). We further added Global-SAN [34] as the second baseline. For a fair comparison, we set the size of the first three coding layers of Global-SAN and GSDAN as the same as GEDAN’s. All these baselines and our GEDAN were trained and tested on 50 by 50 low-resolution facial images. The CED curve results are shown in Fig. 3.8. While GSDAN improves estimation accuracy marginally compared to Global-SAN, our GEDAN consistently outperforms both GSDAN and Global-SAN greatly. To further show the strength of GEDAN, we display the overall NRMSE distribution of these three models on LFPW, HELEN and IBUG in Fig. 3.7. It shows that the NRMSE distribution of GEDAN appears as a narrow peak with smaller errors while the other two are flat-

Table 3.3: Pitch, yaw and roll angle distributions of the 689 testing images

	$0^\circ \sim 10^\circ$	$10^\circ \sim 20^\circ$	$20^\circ \sim 30^\circ$	$\geq 30^\circ$
$\#p^\circ$	442	177	47	23
$\#y^\circ$	331	191	123	44
$\#r^\circ$	499	156	25	9

Table 3.4: NRMSE result (in percentage) of GEDAN and its improvement against the Global-SAN in various rotation groups

	$0^\circ \sim 10^\circ$	$10^\circ \sim 20^\circ$	$20^\circ \sim 30^\circ$	$\geq 30^\circ$
p°	9.43 (+2.02)	10.80 (+2.65)	13.75(+4.53)	24.23 (+2.87)
y°	8.11 (+0.99)	10.41 (+2.06)	13.64 (+4.78)	21.27 (+7.53)
r°	9.01 (+1.80)	12.58 (+3.01)	18.72 (+5.02)	28.22 (+7.25)

tened out. Hence, GEDAN can effectively mitigate the pose and illumination problem and provide a more accurate starting pose.

We then conducted parameter analysis by changing the number of exemplars (K) of GEDAN on the LFPW dataset. Specically, we set K to 0, 10, 20, 50, 80, 120, and recorded the corresponding overall NRMSE values. The results are shown in Table 3.2. It can be seen that the alignment accuracy increases continually with more exemplars. However, after K reaches 50, additional exemplars only improve the result marginally while they largely increase the training time and model size. Hence, we set K to 50 for our global CNN model. To have a comprehensive understanding of the benefits brought by employing exemplars, we further investigated the relative improvement of GEDAN against Global-SAN with various pitch, yaw and roll angles. Specifically, we assembled the 50×50 testing images of LFPW, HELEN and IBUG datasets together (in total we have 689 testing images), and analyzed their pitch, yaw, and roll angle distributions. Let's denote the pitch angle, yaw angle and roll angle as p° , y° and r° respectively. We approximated their distributions by splitting them into four groups according to their

absolute values, *e.g.*, the pitch angles were divided into $0 \leq |p^\circ| < 10^\circ$, $10 \leq |p^\circ| < 20^\circ$, $20 \leq |p^\circ| < 30^\circ$, and $|p^\circ| \geq 30^\circ$. Table 3.3 lists the specific number of images fallen into the respective groups. For example, the first number 442 means there are 442 testing images whose absolute values of pitch angle are within the range of $[0^\circ, 10^\circ]$. Subsequently, we recorded the mean NRMSE of GEDAN and the one of Global-SAN corresponding to each rotation group. The results are shown in Table 3.4. The number pair $9.43(+2.02)$ means the NRMSE of GEDAN is 9.43, and the one of Global-SAN is $9.43 + 2.02 = 11.45$. It can be seen that both the alignment accuracy of GEDAN and the one of Global-SAN degrade as the rotation angle increases. The results also demonstrate that GEDAN consistently outperforms the Global-SAN across all rotation groups. Interestingly, the alignment accuracy improvement increases as the rotation angle increases (except the case corresponding to the group of $|p^\circ| \geq 30^\circ$). The results indicate that the design of GEDAN, *i.e.*, deploying exemplars to assist pose estimation, can effectively improve the alignment robustness against large pose variations.

3.4.3 Evaluation of LDAN

To show the strength of a single LDAN, we created three baselines for comparison. The first baseline had the same structure as LDAN, except that the local output of LAEs was not fed into local regressors. We termed this method as Semi-Localized Deep Auto-encoder Networks (Semi-LDAN). The second baseline regressed on the local patches independently to infer individual landmark positions. Specifically, local features were encoded separately by three layers of LAEs, each of which had 20, 15 and 10 hidden units at the first, second and third layer, respectively². We termed this approach as Extremely Localized Auto-encoder Networks (Ext-LDAN). The third baseline was the Local-SAN [34], which used PCA rather than LAEs to embed local features. The struc-

²The number of hidden units was set in this way such that the overall number of hidden units was comparable to LDAN.

Figure 3.9: The structures of the three baselines and our LDAN. Starting from the top-left one in clockwise order: Semi-LDAN, Ext-LDAN, LDAN and Local-SAN.

tures of these three baselines are depicted in Fig. 3.9. For all these comparing models, we took pose-indexed SIFT features as input, and the predicted poses from GEDAN as their initial poses. The CED curve results are shown in Fig. 3.10. We can make following observations from these results:

- LDAN performs the best among all these single models across three testing datasets.
- LDAN consistently outperforms Semi-LDAN, showing that regressing directly on local features improves detection accuracy.

(a) LFPW

(b) HELEN

(c) IBUG

Figure 3.10: CED curve of a single LDAN on the LFPW, HELEN and IBUG datasets, respectively.

- Semi-LDAN achieves better performance than Local-SAN. This is because Local-SAN utilizes PCA to embed the features in an energy preserving way. On the contrary, LAEs extract pose-informative features from local patches in a task-driven manner.
- Ext-LDAN performs the worst among these single models. This is because Ext-LDAN regresses on local patches independently without considering the global facial structure.

3.4.4 Evaluation of Cascaded LDANs

To evaluate the efficacy of cascading LDANs, we examined the performance of our proposed approach with increasing cascade depth (number of LDANs following GEDAN). Fig. 3.11 shows the corresponding alignment results. It can be seen that cascaded LDANs achieve higher alignment accuracy with the increase of the number of LDANs, and the performance improvement gradually decreases as the number of LDANs increases. This is because the performance of the later LDANs largely depends on the performance of their predecessors. For instance, if the first few LDANs fail to detect facial landmarks (*i.e.*, large detection error), it's unlikely that the following LDANs are able to remedy the failure.

3.4.5 Comparison with the State-of-the-Art Methods

In this session, we first assemble GEDAN with three LDANs to form our CDAN. Then we compare CDAN with 7 state-of-the-art face alignment methods, namely DRMF [32], RCPR [33], SDM [8], IFA [62], GN-DPM [43], TCDCN [50] and CFAN [34] to validate the competency of CDAN. DRMF, IFA and GN-DPM are discriminative fitting methods; RCPR and SDM are boosted regression approaches; CFAN and TCDCN are representative deep learning based methods. For RCPR, we re-trained its model with

Figure 3.11: The performance gains brought by cascading LDANs, where “G+1LDAN” represents one GEDAN followed by one LDAN, and so on.

our training data by using the publicly available code. For the rest methods, we directly used their released codes for testing. Bounding boxes were provided according to their preference (if any). Since DRMF only detected 66 points out from 68, we used the same 66 points to evaluate the comparing approaches. Fig. 3.12 shows the alignment results. It can be seen that CDAN, CFAN and TCDCN perform better than (or comparably with) other methods, which shows the advantage of deep learning based face alignment models. Furthermore, our CDAN consistently outperforms CFAN on all these three datasets. This is because our component networks GEDAN and LDAN outperform CFAN’s Global-SAN and Local-SAN respectively. Compared to TCDCN, our CDAN outperforms it on the LFPW and HELEN testing images, but is less robust than TCDCN on the more challenging IBUG dataset. This can be attributed to the fact that TCDCN was trained with auxiliary datasets in addition to AFW, HELEN and LFPW training images, which brought it alignment robustness against challenging faces.

Since the released code of SDM, IFA and GN-DPM only detect 49 landmarks, we compared our CDAN with them and some other approaches with the same 49 landmark configuration. In the experiment, SDM was able to estimate poses from 286 HELEN testing images, 191 LFPW testing images and 46 IBUG testing images. Hence, we only

(a) LFPW

(b) HELEN

(c) IBUG

Figure 3.12: CED curves of various face alignment methods evaluated on 66 facial points.

(a) LFPW

(b) HELEN

(c) IBUG

Figure 3.13: CED curves of various face alignment methods evaluated on 49 facial points.

Figure 3.14: From the top row to the bottom row are face alignment results on HELEN, LFPW, IBUG respectively. Note that pose, expression and illumination variations are larger in IBUG than the other two datasets. The sixth column enclosed in red rectangle shows three failed alignments.

compared the pose estimation results on these images. Fig. 3.13 reports the CED curves of these approaches. It shows that CDAN performs the best among the three. While CFAN achieves higher estimation accuracy than SDM on LFPW and HELEN datasets, it performs poorly on the much more challenging IBUG dataset. On the contrary, thanks to the strength of GEDAN and LDAN, CDAN is more robust against large pose variations.

Fig. 3.14 shows some face alignment results of our CDAN on different datasets. It can be seen that our model is capable of detecting landmarks even with the presence of large facial pose, exaggerated facial expression, and partial occlusion.

3.4.6 Computational Time

Our approach was implemented on the Matlab R2012b platform, and the implementation of pose-indexed SIFT descriptor was provided by [8]. The average face alignment time for running CDAN on a desktop with core-i5 CPU @3.2GHZ was around 20 ms per image (excluding face detection part). Specifically, for the GEDAN, its average time

cost was 4 ms. For the first two LDANs (working on pose-indexed SIFT features), each of them took around 7 ms. For the last LDAN (using pose-indexed raw pixel vector as input), it took around 2 ms. Hence, our CDAN satisfies the real-time requirement.

3.5 Conclusions

We have proposed a Cascaded Deep Auto-encoder Network (CDAN) for face alignment. The CDAN is composed of a GEDAN and several LDANs. The GEDAN employs several representative exemplars chosen from the training images pool to assist the pose initialization. Compared with the Global-SAN, the incorporation of these exemplars enlarges the capability of the GEDAN. Hence, GEDAN consistently achieves higher estimation accuracy across all ranges of poses than the Global-SAN. For the LDAN, it utilizes LAEs instead of PCA to embed the pose-indexed SIFT features. The outputs of LAEs are directly fed to the local regressors for regression. In this way, local features have direct impact on pose estimation. Moreover, due to this direct link, training errors can be readily led to update the parameters of LAEs, thereby driving LAEs to learn more expressive features. Therefore, the LDANs are capable of fine-tuning the landmark positions more delicately than Local-SANs. Experimental results have shown that our approach consistently outperforms state-of-the-art face alignment methods. Furthermore, our CDAN achieves real-time performance with Matlab on a common desktop without customized design for speed up, *e.g.*, parallel computing.

While CDAN has achieved favorable alignment results, it's incapable of detecting landmarks from arbitrarily rotated faces. To this end, we introduce a new face alignment approach based on convolutional neural networks in the following chapter, which can robustly detect face landmarks from faces with arbitrary in-plane rotation, large scale variations and face translation.

Chapter 4

Holistic Face Alignment Using HiCNN

While CDAN works well on near-upright faces, it's incapable of detecting landmarks from arbitrarily rotated facial images. To meet the in-the-wild face alignment requirement, we present a robust and accurate face alignment approach which is able to detect landmarks from facial images with *arbitrary* in-plane rotation and a wide range of scale variations. To this end, we leverage the strength of the Convolutional Neural Networks (CNN) and devise a Hierarchical-CNN (HiCNN) cascade. Specifically, our model consists of a global CNN, a part-based CNN and a patch-based CNN. The global CNN generates a preliminary four-landmark configuration from the low-resolution facial image. Based on this preliminary result, part-based CNN estimates the landmark positions from the corresponding facial parts on a larger resolution. Lastly, the patch-based CNN refines the landmark positions from the view of pose-indexed patches at the highest resolution. All these three models are specially tailored for face alignment. Experimental results on three datasets show that the proposed HiCNN achieves robust and accurate alignment performance with real-time speed.

f

Figure 4.1: A group selfie photo taken in the real-world scenario. The blue points on the faces are the facial landmark detection results obtained by our approach. Note that faces are rotated in a wide range of angles. Moreover, some facial images are blurry and partially occluded.

4.1 Introduction

While current face alignment approaches have achieved promising results, most of them assume the foregoing face detection step produces a consistent tight bounding box, and their performance degrades greatly when the bounding box deviates from the ideal position. Moreover, the majority of the face landmark detection methods (including our CDAN in Chapter 3) target at aligning near upright faces, where faces have small in-plane rotation angle range from the upright poses. Nonetheless, faces can exhibit large pose variations in the real-world scenario, *e.g.*, a group selfie in Fig. 4.1 has 6 faces with rotation angles evenly covering 360° . The state-of-the-art face alignment approaches may fail in dealing with this challenging scene. In this chapter, we explicitly deal with the inconsistency of face detection and large face rotation issues. Specifically, we

Figure 4.2: The pipeline of our HiCNN for facial landmark detection.

present a powerful face alignment approach which is able to deal with *arbitrary* in-plane rotation and large scale variation.

In the literature, several approaches [35, 67] apply deep convolutional neural networks to estimate landmark positions. Sun *et al.* [35] devised a multi-stage Convolutional Neural Networks (CNN) for landmark detection. Specifically, a CNN is deployed to initialize the landmark positions and another two networks are utilized to locally adjust the shape. Similarly, Zhou *et al.* [67] devised a four-level CNN cascade for face alignment. The first two-level CNNs estimate landmarks from the holistic facial image, and the next two-level CNNs refine the landmark positions from part (component) images. While these works share similar philosophy (*e.g.*, coarse to fine, deep networks) with our approach, our HiCNN deals with much more challenging problem, *i.e.*, accurate face landmark detection from facial images with arbitrary in-plane rotation and a wide range of scale variations. The robust alignment performance is made possible with our novel structure designs.

For the pose initialization, we devise a global CNN and a part-based CNN to provide a preliminary face configuration. The global CNN takes the low-resolution holistic facial image as input and generates four-landmark positions. Since the global CNN is to detect landmarks from faces of arbitrary in-plane rotation and inconsistent face bounding boxes, this task is extremely challenging even for a deep CNN model. To ease the

difficulty, we design a multi-channel switch to divide the training samples into several groups, where each group has similar facial poses. In this “divide and conquer” manner, each top channel of the global CNN deals with a much simpler task, *e.g.*, detecting the landmarks from facial images with in-plane rotation between 60° and 90° . With the landmark detection result of our global CNN, we rotate the facial image to the upright position and normalize it to a predefined scale. Then part images are extracted from the four detected facial landmarks correspondingly, which are fed to the part-based CNN to derive the full configuration of facial landmarks as the initial pose. To increase the robustness of the part-based CNN model against occlusion, we add a *detector* layer which automatically detects the informative regions of the part images and suppresses those occluded ones.

In terms of the pose update stage, we design a patch-based CNN to refine the result at the highest image resolution. Specifically, we exploit the fact that neighboring landmarks share similar facial structure and landmarks are unique in their own ways. We thereby devise a local partially sharable filter as the basis for each face landmark. The features from these local filter structures are then fused together at the top fully connected layers to jointly regress the landmark positions. Hence both the features and feature-to-pose mappings are learned in a unified way.

Assembling the global CNN, the part-based CNN, and the patch-based CNN together, our HiCNN is capable of accurately detecting landmarks from facial images with *arbitrary* in-plane rotation, large scale variations and random face shifts¹.

The main contributions of our approach are summarized as follows:

- The proposed HiCNN is able to detect facial landmarks from *arbitrarily* in-plane rotated faces as well as large pose- and scale-variations.
- We propose a global CNN and a part-based CNN to provide a promising starting facial pose for face alignment. In particular, a multi-channel switch is devised to

¹Our runnable code is available upon request.

divide the challenging task into multiple easier tasks for the global CNN, and an occlusion detector layer is employed to filter out occlusion part for the part-based CNN.

- We devise a novel patch-based CNN to better extract pose-informative features from local patches, and to efficiently model the highly nonlinear relationship between local features and face poses.

4.2 Hierarchical Convolutional Neural Network

In this section, we describe the details of our HiCNN model. We first present the overall pipeline of HiCNN, and then introduce the structures as well as the learning processes of the global CNN, the part-based CNN and the patch-based CNN. Lastly, we specify the implementation details.

Fig. 4.2 shows the basic idea of our approach. First, the low-resolution detected facial image (50×50) is fed to the global CNN to obtain a preliminary four-facial-landmark configuration. The facial image is then rotated to the frontal pose and normalized to a predefined scale². Subsequently, we deploy the part-based CNN to generate a full configuration of facial landmarks. We then utilize the patch-based CNN to refine the landmark position. Meanwhile, as the pipeline proceeds, image resolution is gradually increased. Therefore, more details are harnessed to increase the detection accuracy. Lastly, we rotate and scale back the facial pose to align with the original image (“De-normalization”).

Figure 4.3: The architecture of our Global CNN. Our Global CNN consists of three sub networks, namely the Sub-NET, the Sel-NET and the Top-NET. The Sub-NET forms the basis for both the Sel-NET and Top-NET. Letters “C, P, R, S” embedded in the arrows represent a convolutional layer, a max pooling layer, an Rectified Linear Units (ReLU) layer [4] and a softmax layer respectively. *e.g.*, “P,R” denote a max pooling layer followed by an ReLU layer. Each round-corner blue rectangle denotes a feature map group, whose size is specified at its bottom in a form of $w \times h \times c \times N$, where w and h indicates the width and height of a feature map, c and N denotes the number of channels and number of instances in the feature map group. The images on the left-hand side of Sub-NET are some examples of our training data, which span a full range of rotation.

4.2.1 Global CNN

Recall that our global CNN detects landmarks from facial images with arbitrary in-plane rotations and large range of scale variations. This task poses a great difficulty for a deep CNN model. A preliminary experiment (as indicated in Fig. 4.9) shows that a medium-sized CNN model is incompetent for this task. To ease the training difficulty, we build a multi-channel switch for the global CNN. Specifically, the global CNN is composed of a base network termed as Sub-NET, a top network with K regression channels named as Top-NET, and a channel switch termed as Sel-NET. Fig. 4.3 depicts the structure of our global CNN. The Sub-NET extracts the low-level features that are shared to both the

²We normalize an image by aligning its landmarks to a predefined frontal pose first, and then we center-crop the image to 100×100 .

Sel-NET and Top-NET. The Sel-NET processes these low-level features, and divides the incoming training images into K groups, where each group has similar rotation angle and scale. For instance, the images on the right-hand side of the Sel-NET in Fig. 4.3 are some examples of the division result, and images in the same row belong to the same group. After this sample division process, each image is assigned with a weight $\alpha_i \in \{0, 1\}$, where $\sum_{j=1}^K \alpha_j = 1$. α_i being 1 indicates that this image belongs to the i th group and it will be fed to the i th channel in the Top-NET. Therefore, the task faced by each regression channel of the Top-NET reduces to a much easier one, such as “detecting facial landmark positions from facial images with in-plane rotation between 60° and 90° ”. The regression results of the K channels are then fused together by weighted sum, *i.e.*, $\mathbf{y}^{(i)} = \sum_{j=1}^K \mathbf{y}_j^{(i)} \alpha_j^{(i)}$, where $\mathbf{y}_j^{(i)}$ is the regression result of the i th image through the j th channel. Note that if the weight $\alpha_j^{(i)}$ is binary, each image only needs to go through its corresponding regression channel assigned by the Sel-NET. On the other hand, each image has to pass through all regression channels, and the final result is a weighted sum of the K regression results.

In our case, the weight $\alpha_j^{(i)}$ is preferable to be binary instead of being continuous. This brings with two advantages. First, each channel only deals with a group of training images with similar poses instead of all images. Hence, training is much faster and easier. Second, the computational load is much lower during the testing phase as each image only passes through one channel. Therefore, we constrain $\alpha_j^{(i)}$ to be binary, which can be achieved by setting the maximum $\alpha^{(i)}$ of the K weights to be 1, and the rest $\alpha^{(i)}$ to 0. Given N training images in a batch, we formulate the training objective as follows,

$$J_G = \min \frac{1}{2} \sum_{i=1}^N \left\| \sum_{j=1}^K \mathbf{y}_j^{(i)} \alpha_j^{(i)} - \mathbf{l}_i \right\|_2^2, \text{ s.t. } \alpha_j^{(i)} \in \{0, 1\}. \quad (4.1)$$

where $\mathbf{l}_i$ is the ground-truth landmark positions of the i th image. By minimizing Eq. (4.1) with the back propagation process, we are able to train the global CNN model. However,

binary constraint makes training error back propagation infeasible. Instead, we relax the α to be continuous at the beginning. The new training objective is formulated as

$$J_G = \min \frac{1}{2} \sum_{i=1}^N \left\| \sum_{j=1}^K \mathbf{y}_j^{(i)} \alpha_j^{(i)} - \mathbf{l}_i \right\|_2^2 - T \sum_{i,j} \alpha_j^{(i)} \log(\alpha_j^{(i)}), \quad (4.2)$$

where T is the annealing parameter, and $-\sum_{i,j} \alpha_j^{(i)} \log(\alpha_j^{(i)})$ calculates the entropy of α . By increasing the value of T , we can gradually binarize α .

Given the relaxation, the training process based on Eq. (4.2) can be easily trapped in a local minimum. This is because it has received little guidance for the sample division task. Here we add a supervision that images of similar rotation angle and scale shall fall into the same group. The new objective is given by

$$\begin{aligned} J_G = \min & \frac{1}{2} \sum_{i=1}^N \left\| \sum_{j=1}^K \mathbf{y}_j^{(i)} \alpha_j^{(i)} - \mathbf{l}_i \right\|_2^2 - T \sum_{i,j} \alpha_j^{(i)} \log(\alpha_j^{(i)}) \\ & + \frac{\lambda_G}{2} \sum_{j=1}^K \sum_{i=1}^N \alpha_j^{(i)} \left\| \mathbf{y}_j^{(i)} - \bar{\mathbf{y}}_j \right\|_2^2, \end{aligned} \quad (4.3)$$

where $\bar{\mathbf{y}}_j = \frac{\sum_{i=1}^N \alpha_j^{(i)} \mathbf{y}_j^{(i)}}{\sum_i \alpha_j^{(i)}}$ is the mean regression result of the j th channel. The third term in Eq. (4.3) can be viewed as a K-means clustering loss. By minimizing this term, training instances are divided into K clusters (groups), and each cluster (group) shares similar landmark configuration. Therefore, our model is a multi-task joint learning model, where facial landmark detection and face clustering are trained together. Zhang *et al.* [66] proposed a similar multi-task joint learning approach for face alignment. Their model learns facial attributes classification and facial landmark detection jointly with a single CNN model. However, their method requires extra supervised information such as facial expressions, gender and head pose other than landmark locations, while our training procedure doesn't require any of these extra labels. For the training procedure, we gradually decrease the value of λ_G and increase T after each epoch.

Specifically, $\lambda_G = s_\lambda \lambda_G$ and $T = s_T T$, where s_λ and s_T are empirically set to 0.9 and 1.1 respectively. The gradient calculation details are given in the following section.

The four landmarks detected by the global CNN are the 38th, 45th, 30th and the average of 63rd and 67th landmark defined on the IBUG website [3]. The detection of these landmarks can be easily applied for face normalization. Moreover, each landmark approximately locates at the geometric center of its corresponding facial part, *e.g.*, the 38th landmark lies at the center of the left eye part (including the left eye and the left brow). Hence, we can extract pose-indexed facial parts from the 4 landmarks and these facial parts are taken as input for the following part-based CNN. Fig. 4.4 illustrates the four landmarks and their corresponding facial parts.

4.2.1.1 The gradient calculations of the global CNN

In this part, we provide the details of gradient calculation of our global CNN. First we focus on the gradient of $\frac{\partial J_G}{\partial \alpha_j^{(i)}}$, where J_G is given in Eq. (4.3). With the derivative chain rule, we can obtain the following expression,

$$\begin{aligned} \frac{\partial J_G}{\partial \alpha_j^{(i)}} &= \left(\sum_{j=1}^K \mathbf{y}_j^{(i)} \alpha_j^{(i)} - \mathbf{l}_i \right)^T \mathbf{y}_j^{(i)} - T - T \log(\alpha_j^{(i)}) \\ &\quad + \frac{\lambda_G}{2} \|\mathbf{y}_j^{(i)} - \bar{\mathbf{y}}_j\|_2^2 + \lambda_G \sum_{k=1}^N \alpha_j^{(k)} (\bar{\mathbf{y}}_j - \mathbf{y}_j^{(k)})^T \frac{\partial \bar{\mathbf{y}}_j}{\partial \alpha_j^{(i)}}, \end{aligned} \quad (4.4)$$

where $\bar{\mathbf{y}}_j = \frac{\sum_{k=1}^N \alpha_j^{(k)} \mathbf{y}_j^{(k)}}{\sum_k \alpha_j^{(k)}}$ and $\frac{\partial \bar{\mathbf{y}}_j}{\partial \alpha_j^{(i)}}$ is given by

$$\begin{aligned} \frac{\partial \bar{\mathbf{y}}_j}{\partial \alpha_j^{(i)}} &= \frac{\mathbf{y}_j^{(i)} \sum_k \alpha_j^{(k)} - \sum_k \alpha_j^{(k)} \mathbf{y}_j^{(k)}}{(\sum_k \alpha_j^{(k)})^2}, \\ &= \frac{\mathbf{y}_j^{(i)} - \bar{\mathbf{y}}_j}{\sum_k \alpha_j^{(k)}}. \end{aligned} \quad (4.5)$$

Let's denote the term $\sum_{k=1}^N \alpha_j^{(k)} (\bar{\mathbf{y}}_j - \mathbf{y}_j^{(k)})^T \frac{\partial \bar{\mathbf{y}}_j}{\partial \alpha_j^{(i)}}$ in Eq. (4.4) as g , then

$$\begin{aligned}
 g &= \bar{\mathbf{y}}_j^T \sum_k \alpha_j^{(k)} \frac{\partial \bar{\mathbf{y}}_j}{\partial \alpha_j^{(i)}} - \sum_k \alpha_j^{(k)} \mathbf{y}_j^{(k)T} \frac{\partial \bar{\mathbf{y}}_j}{\partial \alpha_j^{(i)}}, \\
 &= \bar{\mathbf{y}}_j^T \sum_k \alpha_j^{(k)} \frac{\mathbf{y}_j^{(i)} - \bar{\mathbf{y}}_j}{\sum_k \alpha_j^{(k)}} - \frac{\sum_{k=1}^N \alpha_j^{(k)} \mathbf{y}_j^{(k)T}}{\sum_k \alpha_j^{(k)}} (\mathbf{y}_j^{(i)} - \bar{\mathbf{y}}_j) \\
 &= \bar{\mathbf{y}}_j^T (\mathbf{y}_j^{(i)} - \bar{\mathbf{y}}_j) - \bar{\mathbf{y}}_j^T (\mathbf{y}_j^{(i)} - \bar{\mathbf{y}}_j) = 0
 \end{aligned} \tag{4.6}$$

Therefore, Eq. (4.4) becomes

$$\begin{aligned}
 \frac{\partial J_G}{\partial \alpha_j^{(i)}} &= \left(\sum_{j=1}^K \mathbf{y}_j^{(i)} \alpha_j^{(i)} - \mathbf{l}_i \right) \mathbf{y}_j^{(i)} - T \\
 &\quad - T \log(\alpha_j^{(i)}) + \frac{\lambda_G}{2} \|\mathbf{y}_j^{(i)} - \bar{\mathbf{y}}_j\|_2^2.
 \end{aligned} \tag{4.7}$$

Similarly, we have

$$\frac{\partial J_G}{\partial \mathbf{y}_j^{(i)}} = \left(\sum_{j=1}^K \mathbf{y}_j^{(i)} \alpha_j^{(i)} - \mathbf{l}_i \right) \alpha_j^{(i)} + \lambda_G \alpha_j^{(i)} (\mathbf{y}_j^{(i)} - \bar{\mathbf{y}}_j). \tag{4.8}$$

4.2.2 Part-based CNN

Having detected the four landmarks by the global CNN, we rotate and normalize the facial image to a predefined frontal pose by using similarity transform. Then we extract four facial part images of size 61×61 whose centers are the four landmarks. For example, the left-eye-brow part (marked in green dashed-rectangle in Fig. 4.4) centers on the 38th landmark and covers all the landmarks located in this region. Additionally, we rescale the normalized image to 61×61 to detect the contour landmarks. After extracting these five facial parts from the normalized facial images, five CNN networks of the part-based CNN are deployed on the corresponding part images to detect the facial landmarks independently. We use the same CNN structure for these networks except for

Figure 4.4: The four landmarks we choose to detect by the global CNN are marked as blue asterisks, which are the 38th, 45th, 30th and the average of the 63rd and 67th landmark. The green, black, blue and red dashed rectangles denote the left-eye-brow part, right-eye-brow part, the nose part, and the mouth part respectively. Note that the actual size of the nose part and the size of mouth part are the same, here we enlarge the red rectangle a bit so as to differentiate it from the blue rectangle for clarity.

their top layers. This is because different parts detect different numbers of landmarks.

The details of the part-based CNN structure is specified in Table 4.1.

Since each part image has a limited field of view, the performance of the part-based CNN model can be plagued by occlusions and cluttered background. The occlusion and background patterns are highly diverse whose distribution cannot be effectively modeled without collecting numerous occluded training images. On the other hand, those non-occluded clean facial regions are relatively structured. By emphasizing the clean regions and suppressing the occluded parts, we can increase the robustness of the part-based

Layer	Type	Filter size	Input size
1	C	$4 \times 4 \times 1 \times 10$	$61 \times 61 \times 1 \times N$
2	P,R	-	$58 \times 58 \times 10 \times N$
3	C	$4 \times 4 \times 10 \times 20$	$29 \times 29 \times 10 \times N$
4	P,R	-	$26 \times 26 \times 20 \times N$
	D	$1 \times 1 \times 20 \times 1$	$13 \times 13 \times 20 \times N$
5	C,R	$4 \times 4 \times 20 \times 20$	$13 \times 13 \times 20 \times N$
6	C,R	$3 \times 3 \times 20 \times 100$	$10 \times 10 \times 20 \times N$
7	C,R	$3 \times 3 \times 100 \times 100$	$8 \times 8 \times 100 \times N$
8	P,R	-	$6 \times 6 \times 100 \times N$
9	C	$3 \times 3 \times 100 \times 2L$	$3 \times 3 \times 100 \times N$

Table 4.1: The detailed structure of a CNN network in the part-based CNN. “C,P,R,D” represent a convolutional layer, a max pooling layer, an ReLU layer and a detector layer respectively. The variable L in the last row is the number of landmarks to detect in this facial part. For instance, the mouth part has 20 landmarks and the nose part has 9 landmarks.

CNN model against occlusion. To this end, we add a *detector* layer to identify those clean component-informative areas and set their corresponding weight close to 1, and associate the rest which are either occluded or uninformative regions with weights close to 0.

A partial view on the network structure of the part-based CNN corresponding with the mouth part image is illustrated in Fig. 4.5, where $X \in \mathbb{R}^{13 \times 13 \times 20}$ denotes the output feature maps of the fourth layer. M represents the weight map obtained by the detector, and M' are the replicate weight maps. In addition, Y denotes the result of element-wise product between X and M' .

The detector layer is composed of a sigmoid operator and a convolutional filter w with receptive field 1×1 . Specifically, let's index X, Y, M, M' by i, j, c , where i, j and c index on the rows, columns and channels respectively. Then $M_{ij} = \sigma(\sum_{c=1}^{20} X_{ij}^{(c)} w_c + b)$, where w_c is the c th parameter of w corresponding with the c th channel, and b is the bias. $\sigma(x) = 1/(1 + e^{(-ax)})$ is the sigmoid function which maps an arbitrary real number into the range of $[0, 1]$, and a controls the steepness of $\sigma(x)$, *i.e.*, the larger a is, the steeper $\sigma(x)$ will be. In our implementation, a is empirically set to 10 through

Figure 4.5: A partial view of our part-based CNN architecture corresponding with the mouth part. The convolutional layer, max-pooling layer and the ReLU layer are represented by “C,P,R” respectively. “D” denotes the detector layer and “E” is an elementwise product operator.

validation experiment. Having obtained M , we replicate it 20 times to get M' . Then we perform elementwise product between X and M' to yield Y , $Y_{ij}^{(c)} = X_{ij}^{(c)} * M'_{ij}^{(c)}$. Y is subsequently taken as the input for the fifth layer of the part-based CNN.

Note that each element in the weight map M approximately corresponds with a 5×5 block of the input part image. $M_{ij} = 1$ indicates that the corresponding block is informative for landmark detection and it's not occluded. To have a vivid idea of how weight map works, we visualize M in Fig. 4.5, where the dark pixels are weights close to 0 and bright ones are weights close to 1. We further illustrate how the weight map corresponds with the input part image I by showing the weighted part image I' in Fig. 4.5. First M is scaled to be the same size of I . Then $I'_{ij} = M_{ij}^{scale} * I_{ij}$. We can see that the occluded face region and the uninformative facial area in I are corresponding with large continuous gray area in I' , which are the area associated with weight zero³.

³The part image has been subtracted by the mean image, where the pixel value range of I is between $[-255, 255]$. Therefore, the region of I' equal to 0 are rendered as gray area instead of black.

For example, the majority part of the occluding hand is associated with weights close to 0, while the nostrils and the mouth corner are corresponding with weight 1. We find this weight map pattern is prevalent in our experiments. Specifically, for the mouth landmark detection, the nostrils, cheek contour, and the mouth region are the most informative areas, which are generally weighted by 1.

To train this detector layer, two intuitive ways emerge as options. We can either collect considerable amount of occluded training images and label the occluded area, or synthetically generate them by overlaying various objects on the clean facial images. The first option demands formidable amount of human labor, and the latter one can achieve limited effect as it's quite difficult to collect diverse occluding objects which share similar illumination conditions with the ones of facial images. Instead of working on the dataset, we formulate the objective function for the part-based CNN model as

$$J_P = \min \frac{1}{2} \sum_{i=1}^N \|l_i - f(I_i)\|_2^2 + \frac{\lambda_P}{2} \sum_{i=1}^N (e^T M_i e - \theta e^T e)^2, \quad (4.9)$$

where $f(I_i)$ is the network overall function that maps the i th part image to its corresponding landmark position and l_i denotes ground-truth landmark positions in the i th facial part. Correspondingly, M_i is the weight map of the i th facial part. e is an all one vector, and θ represents our prior knowledge of the area of component-informative region. Setting θ to 1 enforces all the weights to be 1s, which passes the whole facial region without emphasis. In this case, the whole network is equivalent to a conventional CNN model. On the other hand, setting θ to 0 suppresses all regions of the part image. Either case is undesirable. We empirically set θ as 0.5 according to our validation experiment.

After we run these CNNs on five part images independently, we assemble their results together as the detection result (68 facial landmark positions) of the part-based CNN. Specifically, the output of each CNN corresponds to a particular range of land-

marks, *e.g.*, the CNN based on the left eye part estimates the 18th-22nd (the left brow landmarks), and the 37th-42nd landmarks (the left eye landmarks). Hence, we simply map the output of these CNNs to their corresponding landmark indexes to obtain all 68 landmark positions.

4.2.3 Patch-based CNN

With the initial pose derived by the part-based CNN, we further propose a patch-based CNN to undertake local adjustment for each facial landmark. Fig. 4.6 shows the structure of our patch-based CNN. Our proposed model consists of a Sub-NET and a Top-NET (with a bit abuse of notation). The Sub-NET dedicates to extract pose-informative features from local patches. Specifically, a series of Local Convolutional Channels (LCCs) are deployed independently, and each LCC is mounted to a corresponding pose-indexed facial patch. The outputs of these LCCs are then concatenated and fed to the Top-NET for regression. Our network is devised according to the following three observations.

First, informative pose-indexed features need to be extracted for face alignment. Pose-indexed feature has been widely applied in many previous face alignment methods [6, 8, 34, 37]. The existing methods use random ferns [6, 36] and concatenated SIFT features [8, 34] as pose-indexed features. Random fern extracts local features from sparse locations surrounding facial landmarks, which has limited capacity in describing local patches. Concatenated SIFT features simply joins pose-indexed SIFT features into a single feature vector, which inevitably loses some local information. Moreover, SIFT features are hand-crafted features, it's desirable to learn the features from facial patches directly. By learning features in a data-driven way, we can extract pose-informative features with more expressiveness.

Second, facial landmarks exhibit correlations between each one another. For example, the 10 landmarks in the brow region (the left and right brow) have similar local facial

Figure 4.6: The structure of our patch-based CNN. The facial patches on the bottom row are pose-indexed patches. The filters of convolutional layers are indicated by various patterned square blocks, where square blocks of the same color and pattern represent the same filters. The blue rounded rectangles denote feature maps, whose size are indicated by $w \times h \times c$.

structure and textural appearance. On the other hand, these 10 landmarks are unique in their own ways, *e.g.*, the appearance of the leftmost brow tip and the one of the rightmost brow tip are quite different. By exploiting both the uniqueness of each landmark and the correlation between them, we can further improve the alignment accuracy.

Third, after feature extraction, local features need to be integrated to encode the global facial structure, thereby refraining from arbitrary pose regression. This is because the size (field of view) of each facial patch is very small, and it's unreliable to adjust landmark position simply based on a single facial patch. Moreover, if a local facial

patch is occluded, it'll be highly undependable for detection.

Motivated by the above observations, we propose our patch-based CNN for face alignment. (1) To extract pose-informative features from local patches, a number of Local Convolutional Channels (LCC) are configured to learn landmark-specific features separately. For example, the LCC for the eye corner and that for the nose tip are trained separately to learn different features to reflect semantic information of their landmarks. Conversely, the training pose deviations are back-propagated to adjust the parameters of LCC, driving LCC to extract more pose-correlated local features. (2) To exploit the correlation between facial landmarks, we divide the 68 facial landmarks into 5 groups, where each group shares similar facial structure. These five groups are *the contour* (17 landmarks), *the brows* (10 landmarks), *the eyes* (12 landmarks), *the nose* (9 landmarks) and *the mouth* (20 landmarks) [3]. For each group, we assign several group sharable filters. These group filters are shared among the facial landmarks belonging to the same group. To explore the uniqueness of landmarks, we add various landmark specific filters for each landmark. These filters are devoted to extracting features that are specific to the landmark. For instance, each landmark in the brows group has 3 landmark specific filters⁴ and 3 group sharable filters in the first convolutional layer. Similarly, it has 4 landmark specific filters and 4 group sharable filters in the second convolutional layer. In this way, we are able to exploit the uniqueness of each landmark and the correlation among landmarks in the same group. (3) To preserve facial structural information, the output of LCCs are fused together by concatenation, which is followed by two layers of linear regression to obtain the final landmark adjustments. By doing so, local features can be gradually integrated together to implicitly encode the global facial structure.

Therefore, our patch-based CNN adjusts the landmark position from a local perspective under the global facial structure constraint. Moreover, both feature extraction and the feature-to-pose mapping are learned jointly in a principled way.

⁴By number of filters we mean the number of output channels of the convolutional filter layer. For example, a convolutional filter layer of size $4 \times 4 \times 10 \times 12$ has 12 filters since it has 12 output channels.

4.3 Implementation Details

We use 68 landmarks [3] to annotate each face image, and Fig. 4.20 shows some example images of the annotated landmarks. Our HiCNN consists of a global CNN, a part-based CNN and a patch-based CNN. For the convolutional filters in all these models, we deployed zero padding and set stride to 1. For the max pooling layers, we adopted a 2×2 pooling with stride 2. The weight decay parameter and learning rate were set to 0.0005 and 0.01 respectively. After each epoch, the learning rate was scaled by 0.9. In terms of the global CNN, we had 9 regression channels ($K = 9$). For the rest parameters, we set their values as follows: the initial values of λ_G and T were set to 0.1 and 0.001 respectively; the scaling parameters s_λ and s_T were set to 0.9 and 1.1. In terms of the part based CNN, we set λ_P to 0.1, and θ to 0.5. We built our model by using a public Matlab CNN toolbox MatConvNet [135].

Since our approach is sequential in essence, we are not able to train the Global CNN, Part-based CNN and the Patch-based CNN simultaneously. In practice, we trained the Global CNN first, where we augmented the training images by random in-plane rotation, random scaling and random shift. Having trained the Global CNN, we obtained four landmarks from each facial image, from which facial part images are extracted. These part images are taken as the training images for the Part-based CNN. Similarly, the facial patches extracted around the landmarks estimated by the Part-based CNN are the training patches for the Patch-based CNN. The detailed training procedure for each model is described in the following section.

4.4 Experiments

In this section, we first describe the datasets and experimental settings. Subsequently, we investigate the performance of the global CNN, part-based CNN and the patch-based CNN separately. Lastly, we compare our HiCNN with several state-of-the-art face align-

ment approaches.

4.4.1 Datasets and Experimental Settings

- LFPW [39]: There are 1432 facial images in the original LFPW dataset, which are acquired from the Internet. Each image is labeled with 35 facial landmarks. In the IBUG evaluation, these images are re-labeled with 68 landmarks. Since some URLs of the original LFPW images are invalid, there are only 811 training images and 224 testing images available on the IBUG website [3].
- HELEN [134]: There are 2000 training images and 330 testing images in the HELEN dataset, which are collected under wild (uncontrolled) conditions. While there are 194 landmarks in the original annotation, we only use 68 of them for evaluation in our experiments, where the new annotations are also provided by the IBUG website.
- AFW [7]: AFW dataset is collected from the Internet under uncontrolled conditions, and there are 337 face images in this dataset. Similarly, we use the landmark annotations from the IBUG website, where each image has 68 landmarks.
- IBUG [132]: Compared to LFPW and HELEN datasets, IBUG has larger variations on poses, illumination, and facial expressions. There are in total 135 images in this dataset.
- LFW [136, 137]: LFW is a well-known dataset originally designed for face recognition/verification. Zhu *et al.* [136] devised a landmark transfer approach to automatically detect the landmark positions on LFW facial images. The visually correct subset (12905 images) was added to our training datasets. Note that landmark positions of this dataset are less accurate than the other four, as the other four are labeled manually.

Figure 4.7: Some example samples for the global CNN with their corresponding landmark positions marked as red dots. The samples in the blue rectangle are augmented training images, and the ones in the red rectangle are randomly transformed LFPW testing images.

For all these datasets except LFW, we used bounding boxes downloaded from IBUG website. These bounding boxes tightly enclose facial landmarks. For LFW, we added bounding boxes by ourselves which cover all facial landmarks. To robustly estimate facial landmarks situated near the border, their corresponding neighborhood patches need to be incorporated in the bounding box. Hence we enlarged the bounding boxes by 30% on both the width and height. We trained our global CNN, part-based CNN model with the LFPW training images, HELEN training images, AFW dataset and LFW dataset. Differently, we used LFPW, HELEN and AFW to train patch-based CNN. This is because patch-based CNN aims to fine-tune the landmark positions delicately, where the annotation accuracy of LFW dataset is inadequate. While these training data has large number of training images *per se* (16093 images including LFW), most of the images are upright and centered. To train a practical face alignment model, these training images have to be augmented with random rotation, translation and scale. The details of data augmentation will be shortly described in the following sections. The Normalized

Figure 4.8: The NRMSE results of our global CNN against 9 rotation groups on LFPW testing images.

Root Mean Squared Error (NRMSE) was applied to measure the error between estimated landmark positions and the ground truth, which is defined as the average alignment error normalized by the Euclidean distance between the pupils of eyes [6]. Finally, the Cumulative Error Distribution (CED) curves of NRMSE were employed to quantitatively evaluate the performance of different face alignment methods.

4.4.2 Investigation of the Global CNN

The global CNN is the model situated at the front line dealing with facial images of arbitrary in-plane rotation, large scaling and translations. To train this model, it's crucial to augment the training data to cover all these possible variations. First we randomly rotated each training image 6 times with rotation angle uniformly distributed within the range of $[0^\circ, 359^\circ]$. Then we randomly scaled each rotated image 3 times in the scale

range of $[0.8, 1.2]$. Lastly, we randomly shifted the bounding box 3 times. With data augmentation, we had 869,022 training images. A subset of 2000 images were selected as the validation set to early stop the training process from overfitting. Some of the training images with their ground truth landmarks are shown in Fig. 4.7. Note that it's even quite challenging for humans to identify landmarks from these images.

We first conduct parameter analysis by changing the number of channels (K) of the Top-NET. Specifically, we set K to 2, 4, 9, 16, and recorded the corresponding overall NRMSE values. The results are shown in Table 3.2. It can be seen that the alignment accuracy increases continually with more regression channels. However, after K reaches 9, additional channels only improve the result marginally while they significantly increase the training time and model size. Hence, we set K to 9 for our global CNN model.

To evaluate the effectiveness of the global CNN model, we tested the robustness of our model against various groups of rotation angles on the LFPW testing dataset. First, we randomly rotated each testing sample 9 times, where the i th rotation angle was drawn from the distribution of $[(i - 1) * 40^\circ - 20^\circ, (i - 1) * 40^\circ + 20^\circ]$. Hence, each group spanned 40° of rotation angle range. Note that -20° is equivalent to 340° here. Each image was further transformed by random scaling and translation. Some sample testing images are shown in Fig. 4.7. In total, this augmented LFPW testing dataset had 2016 images.

The NRMSE result of our global CNN against 9 rotation groups are shown in Fig. 4.8. The CED curves of our global CNN are tightly clustered together, which shows that our model achieves similar alignment accuracy across different rotation groups, *i.e.*, our model performs consistently against arbitrary in-plane rotation angles.

To validate the benefit brought by the structure of our global CNN, we added three baseline models for comparison. The first two models were conventional deep CNN models, each of which was composed of a Sub-NET and a Top-NET. Their Sub-NETs

Figure 4.9: The NRMSE results of comparing approaches on the augmented LFPW testing images.

K	2	4	9	16
NRMSE(%)	15.16	10.96	8.15	8.10

Table 4.2: The overall NRMSE of the global CNN with various number of regression channels (K).

had the same configuration as the one of Sub-NET in our global CNN. In terms of their Top-NETs, the Top-NET of the first baseline model had the same network structure⁵ and filter size as a regression channel in the global NET. This baseline was termed as *slim CNN*. For the second baseline, its Top-NET had the same structure but different filter size. The filter size was set such that its Top-NET had comparable number of parameters with the one of the Top-NET in the global CNN. This model was termed as *wide CNN*, simply because it had more parameters than *slim CNN*. The details of the convolutional

⁵By network structure we mean the arrangement of layers, *e.g.*, a model with “C, P, R” has the same network structure with another model with “C, P, R” regardless of their convolutional filter size.

Layer	Type	Filter size (<i>slim CNN</i>)	Filter size (<i>wide CNN</i>)
1	C	$4 \times 4 \times 50 \times 50$	$4 \times 4 \times 50 \times 100$
2	C,R	$3 \times 3 \times 50 \times 50$	$3 \times 3 \times 100 \times 100$
3	C,R	$3 \times 3 \times 50 \times 100$	$3 \times 3 \times 100 \times 300$
4	C,R	$3 \times 3 \times 100 \times 50$	$3 \times 3 \times 300 \times 300$
5	C,R	$2 \times 2 \times 50 \times 25$	$2 \times 2 \times 300 \times 140$
6	C	$1 \times 1 \times 25 \times 8$	$1 \times 1 \times 140 \times 8$

Table 4.3: The detailed structure of *slim CNN*’s Top-NET and wide CNN’s Top-NET. “C,R” represent a convolutional layer, and an ReLU layer respectively.

filter configuration are tabulated in Table 4.3. The total number of parameters of the Top-NET in the global CNN, *slim CNN* and *wide CNN* are 1.41 M, 0.16 M, 1.42 M respectively. We further added GEDAN (proposed in Chapter 3) for comparison. GEDAN was retrained using the same training data and the exemplar number was set to 50.

Fig. 4.9 shows the CED curves of these 4 models on landmark detection from all 2016 testing images (LFPW testing images randomly transformed by 9 times). The results show that the global CNN achieves significantly higher alignment accuracy than the other three, and simply increasing the model size from *slim CNN* to *wide CNN* only improves the accuracy slightly. This is because the training tasks these deep models dealt with were extremely difficult. With the multi-channel switch design, our global CNN was able to *divide and conquer* the task with a series of regression channels, thereby substantially mitigating the training difficulty. We also recorded their running time, the global CNN, *slim CNN*, *wide CNN* and the GEDAN spent 2.76 ms, 2.04 ms, 5.58 ms and 2.24 ms respectively in aligning a single 50×50 image. Note that while the global CNN and *wide CNN* have similar model size, the global CNN is much faster than the *wide CNN*. This speed advantage is achieved by using the multi-channel selection switch, *e.g.*, for each image, only one channel is selected for regression.

Figure 4.10: From the top row to the bottom row are the part images corresponding to facial contour, left-eye, right-eye, nose and the mouth part respectively. Each part image is of the size 61×61 .

4.4.3 Investigation of the part-based CNN

After the face normalization part, the major variations each part-based CNN deals with are translation and occlusion. To train this model, we detected facial landmarks from the original training images (without random rotation) using the global CNN. Then we randomly shifted the detected facial landmarks 4 times for each image and extracted facial parts accordingly. Altogether we collected 60,000 part images for each part. Some example training part images are depicted in Fig. 4.10.

It can be seen that most target components are located at the center of the part images, and each part image also incorporates sufficient contextual information, *e.g.*, the nose part image not only covers the nose component, it also includes eyes and mouth. These contextual information ensures each component part-based CNN achieves robust facial landmark detection. One might argue that by jointly regressing on all these five parts, we can further improve the detection accuracy. To validate this assumption, we added a baseline method termed as *combined part-based CNN*, which extracts pose-

informative features from five part images and jointly regresses on the concatenation of these features to obtain landmark positions. Specifically, it consists of a Sub-NET and a Top-NET. The Sub-NET has five channels installed on five part images respectively. Each channel has the same network structure and filter sizes as the first 6 layers of a part-based CNN. The output feature maps of these five channels are concatenated, and subsequently jointly regressed by the Top-NET. In terms of the Top-NET, it is composed of a convolutional layer whose filter size is $3 \times 3 \times 400 \times 400$, followed by “R, P, R” layers, and a convolutional layer with filter size $3 \times 3 \times 400 \times 136$. In this way, features from the five facial parts are extracted independently by each channel of the Sub-NET and they are jointly regressed by the Top-NET.

We further added another baseline to investigate the improvement brought by the additional detector layer. Specifically, the detector layer from our part-based CNN model is removed. Hence all pixels in the part images are treated equally without weighting. This model is termed as *naive part-based CNN*. We tested all these three models on the challenging IBUG dataset. Fig. 4.12 shows the results. Both part-based CNN and its naive version perform much better than the combined version. This is because for the *combined part-based CNN*, the benefit brought by inclusion of all five part images in a single model is quite limited as each part image has covered sufficient facial area for its component detection. Meanwhile, regressing together actually confuses information from these five parts. Differently, the original part-based CNN model maps directly from a part image to its corresponding landmarks. Fig. 4.12 also shows that the additional detector layer improves the alignment accuracy. Some example occlusion detecting results are shown in Fig. 4.13. It can be seen that the detector layer can identify the clean component-informative region from the occluded area. Specifically, the nasolabial folds, nostrils, and cheek contour are some component-informative regions for the mouth. By suppressing unseen occlusion patterns and unrelated background regions, the alignment robustness of the part-based CNN model can be improved.

Figure 4.11: The CED curves of three patch-based CNN models (built from 3 part-based CNN results) on the HELEN testing dataset.

Since the part-based CNN model serves as a predecessor for the patch-based CNN, its effectiveness shall be validated together with the patch-based CNN. Therefore, we trained 3 patch-based CNNs based on the results of the part-based CNN, combined part-based CNN and the naive part-based CNN, respectively. These 3 patch-based CNNs have the same model configuration and training scheme, except their training data are different. Accordingly, the results of these 3 patch-based CNNs are denoted as patch-based CNN, combined patch-based CNN and the naive patch-based CNN. Fig. 4.11 shows the result. The result is highly consistent with Fig. 4.12, which validates the effectiveness of the proposed part-based CNN.

Figure 4.12: The CED curves of three part-based CNN models on the IBUG testing dataset.

Figure 4.13: Each row corresponds to a sample mouth part. For each row, the image on the left is the original part image; the central one is the weighted part image; and the right one is the corresponding weight map generated from the detector layer.

4.4.4 Investigation of the patch-based CNN

In this section, we evaluate the patch-based CNN. We added two baselines for comparison. Both of these two baseline networks consist of a Sub-NET and a Top-NET. For the first baseline, all convolutional filters are unique for each landmark, *i.e.*, filter sharing within a group is not adopted for this model. This model is termed as *unique patch-based CNN*. In terms of the second baseline, we demand that all filters of the same group in the same layer be the same, *i.e.*, filter sharing policy is strictly enforced in this model. This model is named as *fully shared patch-based CNN*. Therefore, our patch-based CNN network is a middle ground between these two baselines, and we denote it

Figure 4.14: The CED curves of comparing models on HELEN testing images.

as *partially shared patch-based CNN* to differentiate it from the other two baselines. All these three models took the same pose-indexed patches as input, which were extracted from the landmark positions estimated by the part-based CNN. We tested these three models on the 330 HELEN testing images. Fig. 4.14 reports the results, which show that all these three patch-based CNN models improve the alignment accuracy compared to the part-based CNN, and our *partially shared patch-based CNN* model outperforms the other two baselines. This is because the “partially share” policy exploits the fact that facial landmarks are similar with each other within the same group while being unique in their own ways. To have a visual interpretation of the performance difference, we display the learned filters corresponding to the landmark 18 and landmark 19 in Fig. 4.15. Landmark 18 and 19 belong to the same group (the brow part). Therefore, their group-shared filters are the same. For example, landmark 18 and 19 share exactly the same filters in the case of *fully shared patch-based CNN* model. From this figure we can see:

1) the filters learned by the partially shared model are various gabor filter alike kernels which are able to extract low-level features such as edges; 2) The filters of landmark 18 learned by the *unique patch-based CNN* model are close to their random initialized values. Thus, this model fails to learn meaningful filters for landmark 18; 3) The majority filters of the fully shared model are similar to average kernels. For the second observation, we conjecture that the facial patches corresponding to landmark 18 have much larger appearance variance than the ones of the rest landmarks⁶, and the *unique patch-based CNN* model may favor those easier landmarks over the challenging one. In terms of the third observation, it is because the “fully share” policy forms a strong regularizer for the training process, where filters are unable to adapt to different facial patches freely. On the contrary, our partially-share model can effectively learn meaningful filters from all landmarks by exploiting the correlation between landmarks in the same group. Meanwhile, landmark-specific filters are granted with sufficient freedom to adapt to their corresponding facial patches.

4.4.5 Comparison with state-of-the-art methods

In this session, we first assemble the global CNN, the part-based CNN with the patch-based CNN to form our HiCNN. Then we compare HiCNN with 11 state-of-the-art face alignment methods, namely ESR [6], ERT [36], LBF [37], DRMF [32], RCPR [33], SDM [8], IFA [62], GN-DPM [43], CFSS [46], TCDCN [50] and CFAN [34] to validate the competency of our HiCNN. DRMF, IFA and GN-DPM are discriminative fitting methods; ESR, ERT, LBF, RCPR and SDM are boosted regression approaches; CFAN and TCDCN are representative deep learning models. For RCPR, we re-trained its model with our training data by using the publicly available code. In terms of the rest models, we used their released codes and their indicated bounding boxes (if any) to

⁶This is because the landmark 18 corresponds to the leftmost eyebrow tip, which can be easily occluded by hairs, sunglasses, *etc.*.

Figure 4.15: The filters of the first convolutional layer in three comparing patch-based CNN models. Each row of filters correspond to a landmark or several landmarks, whose indexes [3] are on the left of the filters. The dashed red rectangle encloses the landmark-specific filters and the blue one contains the group-shared filters. Model names are indicated at the bottom of the corresponding filters.

reproduce their results. We compared our approach with these state-of-the-art methods on the LFPW testing set, HELEN testing set, and IBUG dataset. Fig. 4.16 shows the alignment results of 68 landmarks with ideal bounding boxes (bounding boxes tightly enclosing facial landmarks). It can be seen that our HiCNN performs better than or comparably with other methods. In particular, our model performs the best on the challenging IBUG dataset.

In the literature, the testing images of LFPW and HELEN are assembled as the *common subset*. The IBUG dataset is taken as the *challenging subset*, and the union of them as the *full set* (689 images in total) [37, 46]. For ESR, ERT and LBF, since their codes are not public available, we list their average NRMSE (reported in the corresponding

(a) LFPW

(b) HELEN

(c) IBUG

Figure 4.16: CED curves of face alignment methods evaluated on 68 facial points with ideal bounding boxes.

(a) LFPW

(b) HELEN

(c) IBUG

Figure 4.17: CED curves of face alignment methods evaluated on 49 facial points with ideal bounding boxes.

Method	Common Subset	Challenging Subset	Fullset
ESR [6]	5.28	17.00	7.58
ERT [36]	-	-	6.40
LBF [37]	4.95	11.98	6.32
CFSS [46]	5.13	9.58	6.00
HiCNN	5.05	9.10	5.84

Table 4.4: Comparison of averaged errors with state-of-the-art methods.

papers) in Table 4.4. Our HiCNN is slightly inferior to LBF with a margin of 0.1 on the common subset, but it performs much better than LBF on the challenging subset. Compared to CFSS, HiCNN achieves favorable accuracy on both the common subset and the challenging subset.

Since the released code of SDM [8] and GN-DPM [43] only detect 49 landmarks, we compared our HiCNN with SDM, GN-DPM and some other approaches with the same 49 landmark configuration. In the experiment, SDM was able to estimate poses from 286 HELEN test images, 191 LFPW test images and 46 IBUG testing images. Hence, we only compared the corresponding pose estimation results on these images. Fig. 4.17 reports the CED curves of these approaches. It can be seen that the results on 49 landmarks are consistent with the ones on 68 landmarks, and our approach outperforms the rest methods on the challenging IBUG dataset.

Note that all these results were based on the condition that the provided bounding boxes were ideal (*e.g.*, tightly enclosing the landmarks). To mimic the realistic face alignment scenario, we randomly scaled the bounding boxes in the range of [0.8, 1.2] and shifted the bounding boxes to test the robustness of comparing algorithms. We selected 3 top performers, namely CFSS, TCDCN, and CFAN to compete with our HiCNN. We further added our CDAN (proposed in Chapter 3) for comparison. The results are reported in Fig. 4.18. Note the performance degrades significantly for CFSS, TCDCN, CFAN and CDAN across these 3 datasets compared to the alignment accuracy with ideal bounding boxes, while the alignment accuracy of our approach remains un-

(a) LFPW

(b) HELEN

(c) IBUG

Figure 4.18: CED curves of face alignment methods evaluated on 68 facial points with challenging bounding boxes.

(a) LFPW

(b) HELEN

(c) IBUG

Figure 4.19: CED curves of various face alignment methods evaluated on 68 facial points with arbitrary rotation.

changed on LFPW and HELEN datasets. In terms of the IBUG dataset, the accuracy of HiCNN drops by 10% when NRMSE is equal to 0.1, compared to CFSS's 30% and TCDCN's 35% accuracy degradation. Some example alignment results with realistic bounding boxes are shown in the first two rows in Fig. 4.20. Note even with the presence of scale change, translation, occlusion, and extreme illumination, our HiCNN can still accurately detect facial landmarks.

To showcase the strength of our approach against in-plane rotation, we randomly rotated each testing image with an arbitrary rotation angle. This further increased the alignment difficulty. Similarly, we compared HiCNN with CFSS, TCDCN, CFAN and CDAN. Fig. 4.19 shows the results. It's unsurprising that HiCNN outperforms the rest methods significantly. This is because the HiCNN explicitly deals with arbitrary in-plane rotation. Some example alignment results on arbitrarily rotated faces are shown in the bottom two rows in Fig. 4.20. It can be seen that some facial landmarks are out of image boundary after random rotation, and some are under extreme illumination as well as exaggerated facial expressions. Fig. 4.21 shows some failure cases, we found that: if the occlusion and exaggerated expressions are coupled, the detection result may be unsatisfactory; if the facial pose is rarely seen in the training images, the result may be poor, *e.g.*, the second image; if the illumination condition is extremely side directed, the Sel-NET of the Global CNN could be erroneous (this is the case of the third facial image, the detected facial pose is completely wrong).

4.4.6 Computational Time

Our approach was implemented on the Matlab R2012b platform using the MatConvNET toolbox. The average face alignment time for running HiCNN on a desktop with core-i5 CPU @3.2GHZ was around 20 ms per image (excluding face detection part). Hence, our HiCNN approach satisfies the real-time requirement.

Figure 4.20: Some example facial landmark detection results on IBUG dataset. The first two row of images are facial images cropped from challenging bounding boxes, and the bottom two rows are images with arbitrary random rotation angles coupled with random scale as well as translation.

4.5 Conclusions

We have proposed a powerful face alignment approach named HiCNN. The HiCNN consists of a Global CNN, a part-based CNN and a patch based CNN. The Global CNN mainly addresses the problem of inaccurate face detection result and large pose variations. This pose estimation problem is highly nonconvex, and the conventional deep

Figure 4.21: Some example failure facial landmark detection results on IBUG dataset.

models suffer from getting trapped in the local minimum, resulting in inaccurate landmark detection. To mitigate the problem, the Global CNN adopts a multi-channel switch structure to split the training samples into various clusters, where each cluster shares similar facial poses. Each regression channel thus deals with a much easier training task. This “divide and conquer” design enables the Global CNN to accurately recover the facial landmarks even facing with inaccurate bounding boxes and large pose variations. Compared to the methods based on mean-shape initialization, or those based on conventional SAN and CNN, the Global CNN is much more robust.

With the regression result of the Global CNN, the input faces can be normalized to the predefined frontal pose with similarity transform. In this way, the face variations are substantially reduced. The part-based CNN estimates the full configuration of landmarks from the part images. Since the part image has a limited field of view, it can be plagued by occlusion. The incorporation of the detector layer successfully separates the component informative regions from the occluded or irrelevant background, thereby increasing part-based CNN’s robustness. The patch-based CNN exploits both the correlation and uniqueness among the facial landmarks, which benefits the training process to learn more expressive pose-informative local features to refine landmark positions.

Experimental results have shown that our HiCNN performs better than or comparably with the state-of-the-art face alignment methods in the case of ideal face detection. And HiCNN achieves much higher alignment accuracy than competing methods with challenging bounding boxes. Moreover, HiCNN achieves real-time performance with Matlab on a common desktop.

While HiCNN has achieved promising results on holistic faces, it’s unable to robustly detect facial landmarks from partial faces. In the next chapter, we propose a partial face alignment and recognition approach RPSM, which is capable of aligning arbitrary partial face patch to a holistic gallery image, and subsequently verifying if they belong to the same identity.

Chapter 5

Partial Face Alignment and Recognition

In the previous two chapters, we have introduced two robust and accurate holistic face alignment approaches. With the accurate facial landmark detection, numerous holistic face recognition approaches can be applied. However, in the real-world scenarios, human faces might be occluded by other objects, where holistic face alignment approaches may fail. Hence, it's desirable to devise a robust partial face alignment and recognition approach. While this partial face analysis problem is prevalent in practice, few approaches have been proposed in the literature. In this chapter, we propose a Robust Point Set Matching (RPSM) to automatically align a partial face patch to gallery faces and subsequently identify it. In particular, RPSM's point set matching scheme achieves partial face alignment by taking both the geometric and textural similarity between two feature sets into consideration, and it relies on a point set similarity metric to measure the distance between the matching faces. Extensive experiments on four public face datasets show that RPSM achieves superior results both on partial face alignment and partial face recognition.

Figure 5.1: Several partial face examples. (a) Three partial face patches (in the red ellipse) are from the LFW dataset [1] which are occluded by heads; (b) Face occluded by sunglasses. (c) Arbitrary partial face patch.

5.1 Introduction

Various face recognition approaches have been proposed during the last three decades [13, 14, 16]. These approaches have achieved impressive results on some public databases. However, most of them were designed for controlled scenarios, where all faces have to be pre-aligned to eye positions and normalized to the same size before recognition. Nevertheless, due to the explosion of pervasive multimedia data, it would be onerous, if not impossible, to pre-process these enormous data manually. Recently, the automatic face alignment [10] problem has been extensively studied, and we have presented two robust deep learning based approaches. These landmark detection approaches can accurately align the probe faces to the canonical frontal pose. However, they may suffer from the following two scenarios. First, under uncontrolled environment, facial images can be occluded by other objects. While some of the landmark detection approaches are dedicated to this occlusion problem [33, 51, 52], they only work well on near-frontal faces. Second, face detection result can be inaccurate, where the bounding box may crop out an arbitrary facial patch. Both scenarios generate partial faces, and Fig. 5.1 shows some examples. These partial faces pose great challenge to the current state-of-the-art face alignment approaches. In particular, our experiments in this chapter show that the HiCNN performs unsatisfactory in dealing with partial faces. If the preprocessing face alignment step is inaccurate, the following recognition performance may degrade.

Figure 5.2: Our proposed partial face alignment framework. (a) Feature extraction: keypoints detected by SIFT keypoint detector are marked out as green dots on both images. The left image is the probe partial face image, and the right one is the gallery face image. (b) Keypoint selection by Lowe’s matching scheme: roughly matched keypoints of these two images are connected by green lines, while two pairs of impostor matches are linked by red lines. (c) Point set matching procedure: point set of the probe image marked out as blue diamonds are iteratively aligned to the red-marked point set of gallery image by using RPSM. (d) Matching result: the left one is the warped image using the transformation parameters derived from the matching process, and the right one is the gallery image. Through RPSM, the probe partial face is successfully aligned to the gallery image.

Hence a practical partial face alignment and recognition algorithm is of great demand in dealing with realistic face recognition problems. The proposed approach is desirable to align probe partial faces automatically to the gallery faces, and be robust to occlusions, variations of illumination and pose, *etc.*.

To recognize a probe partial face (either an occluded partial face or an arbitrary facial patch), it’s vital to align it to a holistic/partial gallery facial images accurately, and to devise an appropriate similarity measure to compute their similarity. For example, the occluded facial region should be excluded from the face similarity measurement. To achieve this, an intuitive idea is to first detect facial landmarks [6, 7, 8] in both the gallery and probe images, and then align them with the detected landmarks as well as remove the occluded face parts. However, it remains an open problem for facial landmark detection from arbitrary facial patch. In this chapter, we propose a partial face alignment and recognition approach named Robust Point Set Matching (RPSM). In particular, the partial face alignment is cast as a feature set matching problem, where both the geometric and textural information of the extracted local features are considered. For the partial face recognition, a point-set distance metric is devised to compute the similarity of the partial probe patch and the gallery image over the detected facial feature points.

The matching objective of RPSM is formulated in a linear programming framework, where affine matrix constraint is applied to restrain from unrealistic face warping. Extensive partial face alignment and recognition experiments on four benchmarks verify the effectiveness of our approach.

The rest of this chapter is organized as follows. Section 5.2 introduces the proposed RPSM approach. Section 5.3 provides the experimental results, and Section 5.4 concludes this chapter.

5.2 Robust Point Set Matching

We propose to use local features instead of holistic features. In particular, we apply the Scale-Invariant Feature Transform (SIFT) [138] feature detector to detect local keypoints, which are then described by the SIFT descriptor, Speeded Up Robust Features (SURF) [139] and scale-invariant Local Binary Patterns (SILBP). On matching two keypoint sets, keypoint selection is performed to filter out obvious outliers. The selected keypoint sets are then matched by the proposed RPSM approach, where two feature sets are aligned based on the geometric distribution and the textural information. After alignment, a point set correspondence matrix is derived to indicate the matching correspondence between the matching point sets, and a non-affine transformation function is obtained to register between geometric distributions of these matched keypoints. With the matching result, a point set distance metric is devised to describe the difference between two faces, where the lowest matching distance achieved would be reckoned as sharing the same identity. The face matching process is illustrated in Fig. 5.2. Throughout the rest of the chapter, the matrix transposition is denoted by an apostrophe mark $'$.

5.2.1 Feature Extraction

Since there exist rotation, translation, scaling and even occlusions between the probe image and gallery images of the same identity, it is very difficult to normalize them to eye positions. Without proper face alignment, holistic features would fail to work. Hence, we propose to use local features. First, we detect keypoints with SIFT feature detector. Normally for a typical 128×128 facial image, SIFT feature detector can output hundreds of feature points. The geometric information of each keypoint, which is its relative position in the image frame, is recorded in the geometric feature g .

To describe the local patches of these detected keypoints, we combine the strength of SIFT and SURF keypoint descriptor by simple concatenation, and the dimension of this augmented textural feature is 192. SURF keypoint descriptor is introduced as a complement to SIFT for its greater robustness against illumination variations [140]. While this augmented textural feature is robust against in-plane rotation, scale and illumination, they are originally designed for generic object recognition. To capture more details of facial textures and to accommodate the scaling issue, we incorporate the Scale Invariant LBP (SILBP) [141]. Specifically, uniform rotation invariant version of LBP operator $LBP_{P,R}^{riu2}$ [85] is applied. In terms of the values of P , R , we deploy 4 different sets of values, namely $\{P = 8, R = 1\}$, $\{P = 8, R = 2\}$, $\{P = 16, R = 2\}$, and $\{P = 16, R = 3\}$. These LBP operators are applied on the Gaussian blurred image patches with a specified scale to achieve scale invariance. The features generated by each $LBP_{P,R}^{riu2}$ are then sampled into a corresponding LBP histogram. Hence, for each keypoint, we obtain four scale-invariant LBP (SILBP) histograms, one SIFT histogram, and one SURF histogram. These histograms are concatenated into a single descriptor which is simply termed as “SiftSurfSILBP”.

5.2.2 Keypoint Selection

Having detected the keypoints, we select a subset of keypoints to facilitate the matching process. This is because the number of keypoints of facial image could be up to hundreds. Matching point sets at this scale is computationally intensive. Moreover, irrelevant keypoints (outliers) can mislead the matching process to a local minimum, especially when the number of genuine matching pairs is small compared to the number of impostor pairs. Hence, it's desirable to filter out obvious outliers at the beginning. Here we apply Lowe's matching scheme [138] for keypoint selection, *i.e.*, compare the ratio of distance of the closest neighbor to the one of the second-closest neighbor to a predefined threshold. These coarsely matched keypoint pairs are subsequently fed to our RPSM for finer matching.

5.2.3 Point Set Matching

With the selected keypoints, we have the probe feature set and the gallery feature set corresponding to the probe partial face and the gallery face respectively. Hence, aligning two facial images is equivalent to matching the probe feature set with the gallery feature set. There are three characteristics of this point set matching:

- Subset matching: since the probe image and gallery images are not identical, some keypoints in the probe image may not find their correspondences in the gallery image. Likewise, not all keypoints in gallery images are ensured to be matched. Hence, this point set matching is a subset point matching problem.
- One-to-one point correspondence: this trait is obvious as a keypoint of the nose and a keypoint of an eye in the probe image shouldn't be matched to the same keypoint in the gallery image.
- Non-affine transformation: the appearance of face changes when the perspective or facial expression changes. Such changes, when projected into the 2D image, is

non-affine.

The work of Chui and Rangarajan [109] can meet most requirements listed above. However, its framework only considers feature points' geometric information. In our previous work MLERPM [106], we extended Chui's framework [109] by adding the textural feature matching cost as a regularizer. Nevertheless, we didn't apply any constraints on the affine transformation matrix during the matching process. It is possible that the probe image is largely tilted after matching process, rendering unrealistic facial image warping. To address this problem, we propose a linear programming framework for matching, where constraints of transformation parameters are explicitly enforced.

5.2.3.1 Problem Formulation

Let $h(\cdot)$ be the matching function linking a probe keypoint to a particular gallery keypoint, and $f(\cdot)$ be the non-affine transformation function. The objective is to derive these two functions to minimize the overall matching cost, which is given by

$$J = T(\mathbf{t}^P, h(\mathbf{t}^P)) + \lambda G(f(\mathbf{l}^P), h(\mathbf{l}^P)), \quad (5.1)$$

where $T(\mathbf{t}^P, h(\mathbf{t}^P))$ is the textural matching cost. $G(f(\mathbf{l}^P), h(\mathbf{l}^P))$ measures the geometric matching cost, and λ balances between these two terms.

Matching function $h(\cdot)$: as in [106, 109], matching function $h(\cdot)$ is represented as a binary correspondence matrix $\mathbf{M}$, where $\mathbf{M} \in \{0, 1\}^{N_P \times N_G}$, and $\mathbf{M}_{ij} = 1$ indicates that the i th probe keypoint is matched to the j th gallery keypoint. Since it is a one-to-one matching, there should be at most one 1 in each row of $\mathbf{M}$ and at most one 1 in each column. If there is no 1 in the i th row of $\mathbf{M}$, the i th probe keypoint is an outlier without match. Therefore, the coordinates of $\mathbf{l}_i^P$'s match are

$$h(\mathbf{l}_i^P) = \mathbf{L}^G \mathbf{M}'_{i\cdot}, \quad (5.2)$$

where $\mathbf{M}'_i$ is the i th row of $\mathbf{M}$. Similarly, $\mathbf{L}^G \mathbf{M}'$ records the coordinates of the matched gallery keypoints corresponding to the probe point set $\mathbf{L}^P$.

Textural matching cost: the overall textural matching cost between probe keypoints and gallery keypoints is the summation of textural difference of the corresponding descriptors:

$$T(\mathbf{t}^P, h(\mathbf{t}^P)) = \sum_i^{N_P} \sum_j^{N_G} \mathbf{C}_{ij} \mathbf{M}_{ij}, \quad (5.3)$$

where $\mathbf{C}$ is a pre-computed matching cost matrix and $\mathbf{C}_{ij} = \sqrt{(\mathbf{t}_i^P - \mathbf{t}_j^G)'(\mathbf{t}_i^P - \mathbf{t}_j^G)}$.

Geometrical matching cost: the geometric matching cost is defined as the summation of all the positional distances between geometrically transformed probe keypoints and their matching gallery keypoints:

$$G(f, h) = \sum_i \|f(\mathbf{l}_i^P) - h(\mathbf{l}_i^P)\|_1. \quad (5.4)$$

We select L_1 norm because it could be easily linearized for optimization.

For the non-affine transformation function $f(\mathbf{l})$, we deploy the Thin-Plate Splines (TPS) [142] as our non-affine transformation model. Specifically, TPS aims to minimize the following bending energy

$$E = \sum_i^N \|h(\mathbf{l}_i^P) - f(\mathbf{l}_i^P)\|_2^2 + \lambda \iint \left[\left(\frac{\partial^2 f}{\partial x^2} \right)^2 + \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 f}{\partial y^2} \right)^2 \right] dx dy, \quad (5.5)$$

where $f(\mathbf{l}_i^P)$ is computed as

$$f(\mathbf{l}_i^P) = \mathbf{A} \mathbf{l}_i^P + \mathbf{b} + \mathbf{Q} \phi(i), \quad (5.6)$$

where $\mathbf{A}$ is a 2×2 affine transformation matrix, and $\mathbf{b}$ is a 2×1 translation vector. $\mathbf{Q}$ is a weight matrix associated with ϕ , the latter of which is an $N_P \times 1$ vector recording the internal geometry structure of the probe point set. In particular,

$$\phi(i) = \begin{bmatrix} \|\mathbf{l}_i^P - \mathbf{l}_1^P\|_2^2 \log \|\mathbf{l}_i^P - \mathbf{l}_1^P\|; \\ \vdots \\ \|\mathbf{l}_i^P - \mathbf{l}_{N_P}^P\|_2^2 \log \|\mathbf{l}_i^P - \mathbf{l}_{N_P}^P\| \end{bmatrix}$$

where $\|\mathbf{l}_i - \mathbf{l}_j\|_2^2 \log \|\mathbf{l}_i - \mathbf{l}_j\|$ is the TPS kernel. Combining with Eq. (5.6), Eq. (5.5) can be rewritten as

$$E = \|\mathbf{A}\mathbf{L}^P + \mathbf{b}\mathbf{1}_{N_P}' + \mathbf{Q}\Phi - \mathbf{L}^G\mathbf{M}'\|_2^2 + \lambda \text{tr}(\mathbf{Q}\Phi\mathbf{Q}') \quad (5.7)$$

where $\mathbf{1}_{N_P}$ is an all one vector with N_P elements and Φ is an $N_P \times N_P$ symmetric matrix whose i th column is $\phi(i)$.

In the spirit of linear programming, we further rewrite the above energy minimizing function as our geometric matching function:

$$G = \|\mathbf{A}\mathbf{L}^P + \mathbf{b}\mathbf{1}_{N_P}' + \mathbf{Q}\Phi - \mathbf{L}^G\mathbf{M}'\|_1 + \lambda \|(\mathbf{Q}\Phi)\|_1 \quad (5.8)$$

where $\|(\mathbf{Q}\Phi)\|_1$ calculates the extent of non-rigidity of the transformation. For example, $\|(\mathbf{Q}\Phi)\|_1 = 0$ means that the transformation function $f(\cdot)$ will be fully affine.

Adding together Eq. (5.3) and Eq. (5.8), we have the following overall objective function:

$$\begin{aligned} \min_{\mathbf{A}, \mathbf{Q}, \mathbf{M}, \mathbf{b}} \quad & J = \text{tr}(\mathbf{C}'\mathbf{M}) + \lambda_1 \|\mathbf{A}\mathbf{L}^P + \mathbf{b}\mathbf{1}_{N_P}' + \mathbf{Q}\Phi - \mathbf{L}^G\mathbf{M}'\|_1 \\ & + \lambda_2 \|(\mathbf{Q}\Phi)\|_1 \\ \text{s.t.} \quad & \sum_i \mathbf{M}_{ij} \leq 1, \sum_j \mathbf{M}_{ij} \leq 1, \mathbf{M}_{ij} \in \{0, 1\} \end{aligned} \quad (5.9)$$

The constraints ensure a one-to-one match between a probe keypoint set and a gallery keypoint set.

However, there exist several issues with the above formulation:

- The constraint of $\mathbf{M}_{ij}$ being binary makes the problem NP-hard and cannot be efficiently solved. Therefore, we relax it to be $0 \leq \mathbf{M}_{ij} \leq 1$.
- The optimum of $\{\mathbf{A}, \mathbf{Q}, \mathbf{M}, \mathbf{b}\}$ will be all zero regardless of the values of $\mathbf{L}^P, \mathbf{L}^G, \Phi$. To avoid this, a penalty term of $\mathbf{M}$ is added.
- No constraint is enforced on the affine transformation matrix $\mathbf{A}$, which might skew the whole face into a bizarre shape. We simply regulates the affine transformation be rigid transformation.

Taking the above factors into consideration, we revise Eq. (5.9) as

$$\begin{aligned}
 \min_{\mathbf{A}, \mathbf{Q}, \mathbf{M}, \mathbf{b}} \quad & J = \text{tr}(\mathbf{C}'\mathbf{M}) + \lambda_1 \|\mathbf{A}\mathbf{L}^P + \mathbf{b}\mathbf{1}'_{N_P} + \mathbf{Q}\Phi - \mathbf{L}^G\mathbf{M}'\|_1 \\
 & - \lambda_2 \mathbf{1}'_{N_P} \mathbf{M} \mathbf{1}_{N_G} + \lambda_3 \|\mathbf{Q}\Phi\|_1 \\
 \text{s.t.} \quad & \sum_i \mathbf{M}_{ij} \leq 1, \sum_j \mathbf{M}_{ij} \leq 1, \mathbf{M}_{ij} \geq 0 \\
 & \mathbf{A}_{1,1} = \mathbf{A}_{2,2}, \mathbf{A}_{1,2} = -\mathbf{A}_{2,1}
 \end{aligned} \tag{5.10}$$

In the above formulation, λ_2 controls the number of overall matched pairs. Specifically, a large λ_2 yields a radical matcher which tries to connect all probe keypoints to gallery keypoints, even with the price of mismatch. In this case, many outliers can be included into calculation, rendering inaccurate geometric matching. On the other hand, a small value of λ_2 generates a conservative matcher which connects two keypoints only if they are highly alike. In terms of λ_3 , it controls the energy of non-affine transformation: a large λ_3 limits $f(\cdot)$ to affine transformation, while a small λ_3 encourages the image to transform freely. Therefore, we set λ_3 to a large value in the beginning, where

two matching feature sets are aligned globally with the affine transformation. Then we gradually decrease λ_3 during the matching process to yield a detailed local warping (non-affine transformation).

5.2.3.2 Optimization

We relax our objective function by linearization and re-formulate it as following:

$$\begin{aligned}
\min_{\mathbf{A}, \mathbf{Q}, \mathbf{M}, \mathbf{b}, \mathbf{U}, \mathbf{V}} \quad & J = \text{tr}(\mathbf{C}'\mathbf{M}) + \lambda_1 \mathbf{1}'_2 \mathbf{U} \mathbf{1}_{N_P} \\
& - \lambda_2 \mathbf{1}'_{N_P} \mathbf{M} \mathbf{1}_{N_G} + \lambda_3 \mathbf{1}'_2 \mathbf{V} \mathbf{1}_{N_P} \\
s.t. \quad & -\mathbf{U} \leq \mathbf{A}\mathbf{L}^P + \mathbf{b}\mathbf{1}'_{N_P} + \mathbf{Q}\Phi - \mathbf{L}^G\mathbf{M}' \leq \mathbf{U}, \\
& \sum_{j=1} \mathbf{M}_{ij} \leq 1, \sum_{i=1} \mathbf{M}_{ij} \leq 1, \mathbf{M}_{ij} \geq 0 \\
& -\mathbf{V} \leq \mathbf{Q}\Phi \leq \mathbf{V}, \mathbf{U} \geq \mathbf{0}, \mathbf{V} \geq \mathbf{0}, \\
& \mathbf{A}_{1,1} = \mathbf{A}_{2,2}, \mathbf{A}_{1,2} = -\mathbf{A}_{2,1}
\end{aligned} \tag{5.11}$$

where $\mathbf{U}$ is an auxiliary matrix defining the upper bound of $\mathbf{A}\mathbf{L}^P + \mathbf{b}\mathbf{1}'_{N_P} + \mathbf{Q}\Phi - \mathbf{L}^G\mathbf{M}'$, and $\mathbf{V}$ is the upper bound of $\mathbf{Q}\Phi$. Note that “ $\leq$ ” and “ $\geq$ ” are element-wise operators.

To effectively solve this LP problem (5.11), we employ the successive trust region shrinkage method [126, 143]. To make this thesis self-contained, we briefly describe it as follows. For each probe point $\mathbf{l}_i^P$, we set a corresponding trust region $\mathbf{D}_i$ with its diameter as d in the gallery image. All gallery keypoints outside its trust region are excluded from optimization by setting the corresponding $\mathbf{M}_{ij}$ to 0. Specifically, for a probe keypoint $\mathbf{l}_i^P$, we check all gallery keypoints and set $\mathbf{M}_{ij}$ according to the following criterion:

$$\mathbf{M}_{ij} = 0, \text{ if } E_d(f(\mathbf{l}_i^P), \mathbf{l}_j^G) > 0.5d \tag{5.12}$$

where $E_d(f(\mathbf{l}_i^P), \mathbf{l}_j^G)$ calculates the Euclidean distance between $f(\mathbf{l}_i^P)$ and $\mathbf{l}_j^G$. These newly formed constraints are included in a Constraint Set Ψ for the next iteration. As the

shrinkage process proceeds, each probe point's trust region contracts alongside. Therefore, its matching candidates are fewer and fewer, making the matching results more definite. For example, after the n th iteration, the diameter of trust region would be $d^{(n+1)} = r_1 d^{(n)}$, with $0 < r_1 < 1$. In this way, less gallery points are remained for matching compared to the previous iteration, discretizing correspondence matrix $\mathbf{M}$.

During the trust region shrinkage process, there might occur a scenario where all elements in a row (column) of $\mathbf{M}$ are close to 0. This happens when an outlier has been detected. Note that the keypoint selection part doesn't guarantee the selected keypoint pairs are genuine matching pairs. Some of them can be impostors. The geometric distribution of these impostor pairs lies "unharmoniously" against the rest. Detecting and removing them from the matching process not only accelerates the matching process, but also improves matching accuracy. To detect outliers, we set a threshold τ . If the summation of a row (column) of $\mathbf{M}$ is smaller than τ , that row (column) and its corresponding probe (gallery) keypoint will be removed from the subsequent matching process.

During the matching process, the value of λ_3 is decreased as well: $\lambda_3 = r_2 \lambda_3, 0 < r_2 < 1$, so as to encourage more non-affine transformation. In our implementation, the values of r_1, r_2 and τ are set empirically: $r_1 = 0.8, r_2 = 0.8$, and $\tau = 0.1$. The whole matching process is tabulated as Algorithm 2.

A detailed illustration of RPSM matching process is shown in Fig. 5.3. Note that there exists rotation, translation, scaling, and occlusion between the probe image and the gallery image. In this example, $\lambda_1 = 0.01$. During the first 3 iterations, the alignment is mainly affine transformation since λ_3 is much larger than λ_1 . As matching proceeds, λ_3 gradually decreases. Hence, non-affine transformation plays an increasingly larger role. Finally in Iteration 12, λ_3 is smaller than λ_1 , and it is at this moment the probe keypoints and the gallery keypoints are perfectly aligned. Note that this RPSM matching process is robust to outliers as well. In Iteration 3 and Iteration 8, two pairs of outliers are de-

Algorithm 2: RPSM

Input: $\mathbf{L}^P, \mathbf{L}^G, \mathbf{C}$
Output: $\mathbf{A}, \mathbf{b}, \mathbf{Q}, \mathbf{M}$
 Parameters: $\lambda_1, \lambda_2, \lambda_3, \tau, It_{max}, r_1, r_2$
 Initialize: $d = d_{init}$, Constraint Set $\Psi = \emptyset$
for $It = 1 : It_{max}$ **do**
 Construct Φ ;
 Add Constraint Set Ψ to (5.11) and update $\mathbf{A}, \mathbf{b}, \mathbf{M}$;
 Clear Constraint Set, $\Psi = \emptyset$;
 //Trust region shrinkage
 for each $\mathbf{l}_i^P$ **do**
 Find outsiders $\mathbf{l}_j^G$, where $E_d(\mathbf{l}_i^P, \mathbf{l}_j^G) > 0.5d$;
 Add $M_{ij} = 0$ to Ψ ;
 end
 $d = r_1d, \lambda_3 = r_2\lambda_3$;
 //Outliers detection
 Find outliers $\mathbf{l}_i^P$, where $\sum_j M_{ij} < \tau$;
 Remove $\mathbf{l}_i^P, \mathbf{M}_{i\cdot}$ from $\mathbf{L}^P$ and $\mathbf{M}$ respectively;
 Find outliers $\mathbf{l}_j^G$, where $\sum_i M_{ij} < \tau$;
 Remove $\mathbf{l}_j^G, \mathbf{M}_{\cdot j}$ from $\mathbf{L}^G$ and $\mathbf{M}$ respectively;
end
 Binarize $\mathbf{M}$;
return $\mathbf{A}, \mathbf{b}, \mathbf{Q}, \mathbf{M}$

tected, and they are excluded from the following matching process. More face alignment examples can be found in Fig. 5.18. Our RPSM successfully aligns these partial faces to their corresponding neutral faces. Note that these partial faces are randomly rotated, scaled and cropped. Moreover, some of them are occluded by sunglasses or scarf, and some are with exaggerated expressions.

Probe Image

Gallery Image

Warped Image

Figure 5.3: Illustration of an RPSM matching process. RPSM successfully aligns the input probe image (bottom left) to the gallery image (bottom center), and the result is a warped probe image (bottom right). Six iterations of this RPSM matching process are depicted in the upper red rectangle, where each iteration is enclosed in a light blue square with its corresponding iteration number It and λ_3 indicated at the bottom. Within each iteration, the blue diamonds represent probe keypoints and red crosses represent gallery keypoints. As the matching process proceeds, probe keypoints are gradually aligned to gallery keypoints. Finally, they're perfectly matched in Iteration 12.

5.2.4 Point-Set Distance

Having obtained the transformation parameters between probe and gallery feature sets, we define point set distance metric d_R of two facial images as

$$\bar{d} = \frac{\sum_{i,j} \mathbf{M}_{ij} (\lambda_1 \|f(\mathbf{l}_i^P) - \mathbf{l}_j^G\|_1 + \mathbf{C}_{ij}) + \lambda_3 \|\mathbf{Q}\Phi\|_1}{\sum_{i,j} \mathbf{M}_{ij}}$$

$$d_R = \frac{\bar{d}}{\sum_{i,j} \mathbf{M}_{ij}} \quad (5.13)$$

where $\bar{d}$ calculates the average difference between matched keypoints.

The point set distance defined above is proportional to the average matching difference, and it's inversely proportional to the number of matched point pairs. This distance metric has an intuitive interpretation. The number of matched point pairs indicates the area of two faces which are alike, and the average matching difference indicates the average resemblance of two faces share. With point set distance at hand, RPSM can be directly applied to partial face verification and partial face recognition scenarios effortlessly.

5.2.5 Parameter Settings

In this section, we'll explore the parameter settings of λ_1 , λ_2 and λ_3 . For simplicity, we fixed λ_3 to a constant at first and focused on λ_1 and λ_2 . Intuitively, the value of λ_1 affects the balance between textural matching and geometric matching, *i.e.*, a large λ_1 makes the matching be dominated by geometric matching, while a small λ_1 renders the matching a texture-based matching. Both these two cases are unfavorable. With proper setting of λ_1 and λ_2 , the matching scheme shall be able to detect outliers robustly. For instance, probe point $\mathbf{l}_i^P$ and gallery point $\mathbf{l}_j^G$ are paired up during the matching process, and $\mathbf{M}_{ij} = 1$. However, their positional difference is too large to be compensated by the transformation function, *i.e.*, $\|\mathbf{A}\mathbf{l}_i^P + \mathbf{b} + \mathbf{Q}\phi(\mathbf{l}_i^P) - \mathbf{l}_j^G\|$ is large. In this case, the matcher should be vigilant enough to detect them and set $\mathbf{M}_{ij}$ to 0. To achieve this goal, matching cost of setting $\mathbf{M}_{ij}$ to 0 should be smaller than the one induced by setting $\mathbf{M}_{ij}$ to 1:

$$\begin{aligned} \lambda_1 \|\mathbf{A}\mathbf{l}_i^P + \mathbf{b} + \mathbf{Q}\phi(\mathbf{l}_i^P) - \mathbf{l}_j^G\|_1 + \mathbf{C}_{ij} - \lambda_2 > \\ \lambda_1 \|\mathbf{A}\mathbf{l}_i^P + \mathbf{b} + \mathbf{Q}\phi(\mathbf{l}_i^P)\|_1 \end{aligned} \quad (5.14)$$

Table 5.1: Classification accuracy corresponding to various values of λ_1

λ_1	0.001	0.002	0.005	0.01	0.03	0.06	0.08	0.1	0.2	0.5
u	0.66	0.68	0.72	0.76	0.75	0.73	0.72	0.71	0.68	0.66

Figure 5.4: ROC curves corresponding to various values of λ_1 . The topmost red curve corresponds to $\lambda_1 = 0.01$, and the bottom black one belongs to $\lambda_1 = 0.1$.

One feasible way is to set λ_2 to be $\min(C)$, which ensures that $C_{ij} - \lambda_2 \geq 0$. With simple arithmetic arrangement, Eq. (5.14) becomes

$$C_{ij} - \lambda_2 > \quad (5.15)$$

$$\lambda_1 (\| \mathbf{A} \mathbf{l}_i^P + \mathbf{b} + \mathbf{Q} \phi(\mathbf{l}_i^P) \|_1 - \| \mathbf{A} \mathbf{l}_i^P + \mathbf{b} + \mathbf{Q} \phi(\mathbf{l}_i^P) - \mathbf{l}_j^G \|_1)$$

Due to the uncertainty of the distribution of $\mathbf{l}_i^P$ and $\mathbf{l}_j^G$, we leave the setting of value of λ_1 to the following validation process, which will be detailed shortly. In terms of λ_3 , it should be set to a value that in the first few iterations, the energy of non-affine part $\| \mathbf{Q} \phi(\mathbf{l}_i^P) \|_1$ is close to 0. Similarly, the value of λ_3 is set according to the validation

Table 5.2: Classification accuracy corresponding to various values of λ_3

λ_3	0.001	0.01	0.08	0.16	0.32	0.64	1.2	2.4	5	10
$\mathbf{u}$	0.70	0.74	0.79	0.79	0.80	0.79	0.79	0.79	0.80	0.80

performance.

To obtain the valid parameter settings for λ_1 and λ_3 , we conducted face verification on PubFig database¹. The selected development set was directly used without image size normalization. For positive matches, we paired up every two of the five images for each identity. Thus, we had 600 pairs of genuine matches. For impostor pairs, we randomly paired up two images from two subjects to form 600 pairs. Subsequently, we conducted face verification with various values of λ_1 and a constant large value of λ_3 , namely, $\lambda_1 = 0.001, 0.002, 0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07$, and $\lambda_3 = 100000$. The large λ_3 limits the contribution of non-affine transformation, where we can focus on the effect of variation of λ_1 . The ROC curves and mean classification accuracy ($\mathbf{u}$) corresponding to various λ_1 values are shown in Fig. 5.4 and Table 5.1 respectively.

In Fig. 5.4 and Table 5.1, the verification accuracy increases at first as λ_1 is enlarged from 0.001 to 0.01. This is because a small value of λ_1 makes the matching be dominated by texture-based matching. After λ_1 surpasses 0.01, the verification performance gradually decreases. This can be explained by Eq. (5.15), *i.e.*, a large λ_1 makes the right-hand side large, where the outliers couldn't be detected robustly. Overall, the RPSM achieves the best performance when λ_1 is equal to 0.01.

Having set λ_1 to 0.01, we conducted face verification experiment with various values of λ_3 , *i.e.*, $\lambda_3 = 0.001, 0.01, 0.08, 0.16, 0.32, 0.64, 1.2, 2.4, 5, 10$. The ROC curves and classification accuracy corresponding to various λ_3 values are reported in Fig. 5.5 and Table 5.2 respectively. The results show that the verification performance is poor when λ_3 is smaller than λ_1 . This is because small λ_3 leads to unfavorable situations where the non-affine transformation dominates the affine transformation at the beginning. The

¹This database will be described in the Experiment session.

Figure 5.5: ROC curves corresponding to various values of λ_3 . The bottom red curve corresponds to $\lambda_3 = 0.001$, and the bottom green one belongs to $\lambda_3 = 0.01$.

results also show that classification accuracy stabilizes near 0.79 when λ_3 is much larger than λ_1 . Therefore, the performance of matching is not sensitive to the variation of λ_3 if $\lambda_3 > \lambda_1$. Throughout all our experiments, we set $\lambda_1 = 0.01$ and simply set $\lambda_3 = 5$.

5.3 Experiments

In this section, we first conduct experiment to establish parameter settings. We then evaluate the metric of point-set distance and the RPSM's partial face alignment accuracy respectively. Subsequently, we conduct multiple partial face recognition experiments. Finally, we investigate the performance of our RPSM against holistic face recognition approaches with increasing degrees of *partialness*.

5.3.1 Data Sets

LFW: The Labeled Face in the Wild (LFW) dataset [1] contains 13233 labeled faces of 5749 people, in which 1680 people have two or more facial images. Images in this dataset exhibit large appearance variations as they are taken from uncontrolled settings, including variations in scale, viewpoint, lighting condition, background, make-up, dress, expression, color saturation, image resolution, focus, *etc.*. These variations pose a great challenge to the face recognition task.

AR: The AR dataset [144] contains 126 subjects, including 70 males and 56 females, respectively. For each subject, there exist 26 facial pictures taken in two different sessions (each session has 13 facial images). In each session, there are 3 images with different illumination conditions, 4 images with different expressions, and 6 images with different facial disguises (3 images wearing sunglasses and 3 images wearing scarf, respectively).

EYB: EYB database consists of 2414 frontal facial images of 38 identities photographed under varying controlled illuminations. The public available cropped Yale database is used directly, whose image size is 192×168 .

PubFig: This dataset has 58,797 images of 200 subjects obtained from highly uncontrolled real-world conditions, wherein images differ with each other in terms of pose, illuminations, facial expressions and scene background *etc.*. For performance evaluation, photos of 140 subjects are used as the evaluation set and photos of the rest 60 subjects are taken as the development set. To begin with, there are several issues to be addressed. First, some images are actually duplicates of each other, differing only in image size. These near-duplicate images make face verification much easier. Second, some URL links are unavailable, and these links need to be picked out manually. Third, some images are not of the subject as designated. To deal with these problems, for both the development set and evaluation set, we manually selected 5 images for each identity, where each of them differs from each other in picture size, pose, illumination,

expression, *etc.*. Some sample images of our choice are displayed on the top row in Fig. 5.11.

5.3.2 Baseline algorithms

Our algorithm recognizes partial faces directly without manual face alignment. Thus, we selected the approaches of the same capacity for comparison. In particular, we chose our previous approach MLERPM [106] and Liao *et al.*'s MKD-SRC-GTP [5] (we thank the authors for providing the code).

Furthermore, we designed two new partial face recognition approaches for comparison. The first one was based on Li *et al.*'s work [126]. We simply termed this new approach as Locally Affine Invariant Robust Point set Matching (LAIRPM). In terms of feature extraction, LAIRPM utilized the same features as our SiftSurfSILBP. For feature set matching, local features were matched by Li *et al.*'s matching approach [126]. Its matching objective is given by

$$\begin{aligned} \min_{\mathbf{M}} \quad & J = \text{tr}(\mathbf{C}'\mathbf{M}) + \lambda_1 \|(\mathbf{I} - \mathbf{W})\mathbf{M}\mathbf{L}^{G'}\|_1 - \lambda_2 \mathbf{1}'_{N_P} \mathbf{M} \mathbf{1}_{N_G}, \\ \text{s.t.} \quad & \sum_j \mathbf{M}_{ij} \leq 1, \sum_i \mathbf{M}_{ij} \leq 1, \mathbf{M}_{ij} \geq 0 \end{aligned} \quad (5.16)$$

where $\mathbf{W}$ is a matrix recording the local structure of the probe feature set, and it's derived by reconstructing each feature point with its local neighborhood. Neighborhood is built either through Delaunay Triangulation (DT) or k-nearest-neighbor (kNN) [126]. We tried both schemes and found kNN with $k = 5$ achieved the best result. In terms of λ_1 and λ_2 , LAIRPM employed the same setting as in our RPSM (detailed in Section 5.2.5), where $\lambda_1 = 0.01$ and $\lambda_2 = \max(\mathbf{C})$. After feature matching, the matching

Figure 5.6: Five example facial pairs generated from AR dataset with their landmarks annotated in red dots. The images in the first row are holistic faces with neutral expression and illumination, and the ones in the second row are their corresponding partial faces.

distance metric d_L of LAIRPM is computed as

$$\bar{d} = \frac{\text{tr}(\mathbf{C}'\mathbf{M}) + \lambda_1 \|(\mathbf{I} - \mathbf{W})\mathbf{M}\mathbf{L}^{G'}\|_1}{\sum_{i,j} \mathbf{M}_{ij}}$$

$$d_L = \frac{\bar{d}}{\sum_{i,j} \mathbf{M}_{ij}} \quad (5.17)$$

where $\bar{d}$ calculates the average matching distance between two feature sets.

The second approach was built upon CPD [145]. It's non-trivial to extend CPD to incorporate textural consistency in matching. Hence we directly deployed its released code to match feature sets based on the geometric information. The non-rigid RBF kernel was employed for matching. In terms of the matching distance, it was simply set as the average Euclidean distance between the matched feature points. In the following experiments, unless otherwise specified, CPD, LAIRPM and MLERPM utilized SiftSurfSILBP as their local features.

5.3.3 Evaluation of the Point-Set Distance Metric

To evaluate the proposed point-set distance metric d_R in Eq. (5.13), we designed four baseline metrics:

Table 5.3: Face verification accuracy corresponding with various distance metrics

Metric	d_R	d_G	d_T	d_A	d_M
u	0.80	0.65	0.66	0.78	0.44

- (i) Geometric metric d_G : this metric only considers geometric information in calculation. Specifically,

$$d_G = \left(\lambda_1 \sum_{i,j} \mathbf{M}_{ij} \|f(\mathbf{l}_i^P) - \mathbf{l}_j^G\|_1 + \lambda_3 \|\mathbf{Q}\Phi\|_1 \right) / (\sum_{i,j} \mathbf{M}_{ij})^2.$$

- (ii) Textural metric d_T : this metric only considers textural information in calculation.

In particular,

$$d_T = \left(\sum_{i,j} \mathbf{M}_{ij} C_{ij} \right) / (\sum_{i,j} \mathbf{M}_{ij})^2.$$

- (iii) Affine transformation metric d_A : this metric disregards the non-affine portion, *i.e.*,

$$d_A = \left(\sum_{i,j} \mathbf{M}_{ij} \lambda_1 \|f(\mathbf{l}_i^P) - \mathbf{l}_j^G\|_1 + C_{ij} \right) / (\sum_{i,j} \mathbf{M}_{ij})^2.$$

- (iv) Mean matching difference metric d_M : this metric calculates the average matching difference, *i.e.*, d_M is equal to the $\bar{d}$ of Eq. (5.13).

We conducted face verification on the selected development set of the PubFig database. In particular, we had 600 pairs of positive matches, and 600 pairs of impostor matches. The matching process was conducted by the proposed RPSM approach. After matching, the proposed metric d_R and the four baseline metrics were employed to measure the distance between two images. Table 5.3 records the classification accuracy corresponding with these five metrics. The results show that the proposed metric d_R achieves the best verification performance. d_G and d_T performs poorly since both of them only consider information from single modality (either texture or geometry). d_A is inferior to d_R . This is because the non-affine part of matching accounts for the geometric facial structure differences between two faces. Surprisingly, d_M performs the worst. This might be due

to the fact that d_M merely calculates the average similarity between matched regions, and it doesn't measure the area of the matched regions.

5.3.4 Partial Face Alignment Performance

In this section, we evaluate the alignment capability of our approach. Instead of explicitly detecting landmark positions, RPSM aligns two matching images by deriving a non-linear geometric transformation function $f(l)$. Hence, validation of alignment accuracy of RPSM is equivalent to investigating whether $f(l)$ can faithfully recover the ground-truth transformation between these two images. To do so, we generated 600 pairs of images from the AR dataset, where each pair of images belonged to the same identity but taken under different conditions. Specifically, for each pair, one image was a holistic facial image with frontal pose and neutral expression, and the other one was a partial face which was taken under different illuminations, or with exaggerated facial expressions. For the holistic images, we detected 51 landmark positions with our HiCNN (the facial landmark detection approach proposed in Chapter 4). Since these images were mainly with frontal poses and centered in the image, HiCNN can accurately detect facial landmarks. In terms of partial faces, they were generated by randomly transforming holistic facial images. First each image was randomly rotated with rotation angle uniformly distributed in $[-20^\circ, 20^\circ]$. Then the rotated image was randomly cropped to size as $h \times w$, where h and w could be as small as 0.8 times of the original height and width respectively. Lastly, the cropped patch was randomly scaled between 0.8 to 1.2. Since we know the exact value of rotation angle, scale, and shift for each image, we can get the ground-truth landmark positions of each partial face as well. Some example facial image pairs are shown in Fig. 5.6.

We compared our RPSM with a state-of-the-art feature set matching approach CPD [145], as well as four landmark detection approaches which are CFAN [34], SDM [8], CDAN (our facial landmark detection approach proposed in Chapter 3), and HiCNN. We further

added an appearance similarity maximization approach named RASL [2] for comparison. Note that LAIRPM was not compared as it doesn't yield geometric transformation function. For a face pair, let's denote the i th landmark position of the holistic facial image as $\mathbf{l}_i^H$, and the one of the partial face as $\mathbf{l}_i^P$. Given the face pair, the feature set matching approach is able to derive a transformation function $f(\mathbf{l})$, with which we can estimate the landmark positions of the partial face by $f(\mathbf{l}^H)^2$. Then we measure the alignment accuracy of a feature set matching method by

$$\epsilon = \frac{\sum_{i=1}^N \|\mathbf{l}_i^P - f(\mathbf{l}_i^H)\|^2}{dN}, \quad (5.18)$$

where d is the inter-pupil distance of the partial face, and N is the number of landmarks (in this case $N = 51$). Error ϵ is the Normalized Root Mean Squared Error (NRMSE), which is defined as the average alignment error normalized by the inter-pupil distance [6]. For the landmark detection approaches, their alignment accuracy is assessed by $\epsilon = \sum_{i=1} \|\mathbf{l}_i^P - \hat{\mathbf{l}}_i^P\|^2 / dN$, where $\hat{\mathbf{l}}^P$ are their estimated landmark positions. Lastly, the Cumulative Error Distribution (CED) curves of NRMSE were used to quantitatively evaluate the performance.

Fig. 5.7 shows the CED curves results. Feature set matching approaches achieve significantly higher alignment accuracy than the holistic face alignment methods (including the four landmark detection methods SDM, CFAN, CDAN, HiCNN and one appearance similarity maximization algorithm RASL). This is because partial faces are extremely challenging for face alignment methods, where some key facial components (such as brows, mouth) can be missing. On the contrary, feature set matching approaches conduct face alignment by matching two local feature sets. It doesn't matter if any of the facial components are occluded or missing, so long as there exist similar facial patches between the matching faces. Within the feature set matching group, both the

² $\mathbf{l}^H$ is provided as prior knowledge since landmark detection from unoccluded frontal face can be deemed as a solved problem.

Figure 5.7: The CED curves of various comparing approaches on 600 partial faces.

RPSM and MLERPM achieve the better performance than CPD. This is due to the fact that RPSM/MLERPM considers both the geometric consistency and textural similarity during alignment, while CPD only utilizes the geometric information. In terms of MLERPM, it performs poorer than RPSM as its affine-part matching is unregulated, which may result in tilted and sheared facial images.

5.3.5 Partial Face Recognition Performance

5.3.5.1 Partial face verification on LFW dataset

We conducted face verification on LFW dataset, where we applied our algorithm on the original unaligned LFW images directly. We first used the OpenCV implementation of the Viola-Jones face detector to detect and crop out the facial regions of all images. The cropped faces underwent the same random transformation process as in Section 5.3.4. Some sample partial face pairs are shown in Fig. 5.8. Note that in this experiment, both

Figure 5.8: Example facial images from the LFW dataset after random transformation in View 2. Columns in the red box: matched image pairs. Columns in the blue box: mismatched image pairs.

the gallery and probe images were partial faces. In terms of the verification process, we strictly followed the experiment protocol in View 2 outlined on the homepage of LFW dataset. To demonstrate the effectiveness of our SiftSurfSILBP features, we added a group of experiments where comparing algorithms worked on SiftSurf features. These approaches were denoted with postfix “-SiftSurf”. Similarly, the approaches working on SiftSurfSILBP were denoted with postfix “-SiftSurfSILBP”.

To further showcase the effectiveness of the RPSM, we added the High Dimensional LBP (HDLBP) [95] approach for comparison. We implemented HDLBP as follows. First, we utilized HiCNN to detect 25 inner facial landmarks for each image. Subsequently, each image was normalized to the frontal pose and was further resized to 5 scales. For each scale, fixed-size image patches at all facial landmarks were cropped, which were then described by LBP features. Finally these LBP descriptors were concatenated to a 118,000 dimensional feature. This feature was projected by PCA to 750 dimension. For verification, Joint Bayesian [96] was employed. We tested our im-

plementation on the funneled LFW images [11] with the unrestricted setting without outside training data, and it achieved 93.45% accuracy, which was comparable to the reported result (93.18%). To apply HDLBP in this partial face recognition scenario, we first employed HiCNN to detect landmarks from the randomly transformed LFW images. The distance between the matched/mismatched image pairs were then calculated by applying the trained joint bayesian model.

The ROC curves of comparing algorithms are shown in Fig. 5.9, and their classification accuracy (u) as well as their corresponding standard deviation (S_E) are listed in Table 5.4. The results show that RPSM working on SiftSurfSILBP achieves the highest classification accuracy. Moreover, algorithms using SiftSurfSILBP outperform their counterparts on SiftSurf features, which shows that SILBP can extract complementary information for SiftSurf features. HDLBP performs the poorest among the comparing approaches, and its partial face recognition accuracy 49.32% is significantly worse than its holistic face recognition accuracy 93.45%. The performance degradation is mainly due to the incompetency of HiCNN in dealing with partial faces. To our best knowledge, the state-of-the-art landmark detection approaches are unable to estimate facial landmarks robustly from partial faces, especially when one or more facial components are cropped out.

5.3.5.2 Face verification on PubFig dataset

To further validate the effectiveness of our approach on face verification, we conducted face verification experiment on the selected PubFig dataset. The selected evaluation set was directly used in our verification experiment without image size normalization. To begin with, we constructed positive matching pairs and impostor pairs. For positive matches, we paired up every two of the five images for each identity. Therefore, we had 1400 positive matches. For impostor pairs, we randomly paired up two images from two identities to construct 1400 impostor matches for verification. The ROC curves are

Figure 5.9: Verification ROC curves of the comparing algorithms on LFW dataset.

depicted in Fig. 5.10, and the accuracy ($\mathbf{u}$) as well as its standard deviation ($\mathbf{S_E}$) are reported in Table 5.5.

From the ROC curves and classification accuracy, we find that our approach RPSM-SiftSurfSILBP achieves the best verification performance. Algorithms working on SiftSurfSILBP features achieve better performance than their counterparts on SiftSurf features.

5.3.5.3 Partial face identification on arbitrary partial face patch

To demonstrate the strength of our algorithm on recognizing arbitrary partial face patches, we randomly transformed the images of the selected PubFig dataset (evaluation set) to generate partial face patches. The randomization process was as the same as the one de-

Table 5.4: Verification accuracy and corresponding standard deviations of the comparing algorithms on LFW

Method	$\mathbf{u} \pm \mathbf{S_E}$
RPSM-SiftSurfSILBP	0.7165 ± 0.0157
RPSM-SiftSurf	0.7081 ± 0.0146
LAIRPM-SiftSurfSILBP	0.7073 ± 0.0168
LAIRPM-SiftSurf	0.7040 ± 0.0102
MLERPM-SiftSurfSILBP	0.6722 ± 0.0183
MLERPM-SiftSurf	0.6555 ± 0.0153
MKD-SRC-GTP	0.6818 ± 0.0177
CPD-SiftSurfSILBP	0.6162 ± 0.0119
HDLBP	0.4932 ± 0.0109

Table 5.5: Verification accuracy and corresponding standard deviations of the comparing algorithms on PubFig

Method	$\mathbf{u} \pm \mathbf{S_E}$
RPSM-SiftSurfSILBP	0.7889 ± 0.0175
RPSM-SiftSurf	0.7335 ± 0.0240
LAIRPM-SiftSurfSILBP	0.7000 ± 0.0275
LAIRPM-SiftSurf	0.6932 ± 0.0170
MLERPM-SiftSurfSILBP	0.6864 ± 0.0204
MLERPM-SiftSurf	0.6618 ± 0.0202
MKD-SRC-GTP	0.6982 ± 0.0156
CPD-SiftSurfSILBP	0.6146 ± 0.0300

Figure 5.10: Verification ROC curves of the comparing algorithms on PubFig dataset.

tailed in Experiment 5.3.5.1. Some sample partial face images are shown on the bottom row in Fig. 5.11.

With partial faces at hand, we randomly split this transformed dataset into five subsets, where each subset had 140 images with one image per person, and any two subsets didn't share the same images. For testing, we conducted face verification with five-fold testing scheme. Specifically, within each round, one subset was selected as gallery images, and the rest four were taken as probe images. Therefore, in this experiment, both the gallery and probe images were partial faces, which posed great challenge to recognition approaches. To demonstrate the effectiveness of our matching approach, we added three baseline methods in addition to LAIRPM, MKD-SRC-GTP and MLERPM. These three baseline algorithms worked on textural part of local features

Figure 5.11: Samples of our selected PubFig dataset. Top row: original selected images of Adriana Lima. Bottom row: corresponding random transformed images of the top row.

(SiftSurfSILBP) alone:

- The first one applied Lowe’s matching method to match textural feature sets of gallery images and probe images. The number of matching pairs was set as similarity criterion.
- The second one was Hausdorff distance (HausDist) [146] which calculates the largest distance between closures of two textural feature sets.
- The third method was Earth Mover’s Distance (EMD)[147], which measures the minimum cost of transforming one distribution of textural feature set into the other. We set number of K-means clusters to 8, which was the best result achieved across various values of k , ranging from 5 to 13.

Table 5.6 and Fig. 5.12 show the corresponding experimental results. From the results we can make several observations:

- Our method RPSM obtains the best recognition rates at most ranks. Note that it performs consistently better than MLERPM at all ranks, showing the benefits of regulating the affine transformation matrix.

Figure 5.12: Recognition rates of the comparing algorithms at various ranks on PubFig

Table 5.6: Average recognition accuracy (%) of the comparing algorithms at various ranks on PubFig

Method	Rank 1	Rank 10	Rank 20
RPSM	42.86	65.00	74.29
LAIRPM	37.14	64.29	72.86
MKD-SRC-GTP	38.57	62.14	72.14
MLERPM	27.86	52.86	64.29
CPD	28.36	51.93	62.29
Lowe	25.00	49.29	57.86
EMD	3.57	20.00	30.71
HousDist	0.71	10.71	19.29

Figure 5.13: Samples from the AR dataset. (a) Cropped gallery image with neutral expression. (b) Cropped probe images occluded by sunglasses and scarf. (c) Original sized probe image with occlusion.

- (ii) LAIRPM and MKD-SRC-GTP have competitive performance against RPSM at large ranks, *i.e.*, from Rank 15 onwards. But their performance are poorer than RPSM's at the first 10 ranks. Higher recognition rates at these ranks are critical in practical applications.
- (iii) HausDist, EMD and Lowe's matching approaches achieve poor performance. This is because matching only on textural features merely exploits partial information of facial images, and textural features alone are not discriminative enough.

5.3.5.4 Partial face identification under occlusion

We conducted several experiments of “partial face recognition under occlusion” on AR database [144] and on EYB database. For AR, a subset containing 50 male subjects and 50 female subjects was selected from the AR dataset as in [20, 26]. For each identity, the first image in the first session (without occlusion and has neutral expression) was used as gallery image, while six images with sunglasses and six images with scarves from both sessions were used for testing. For fair comparison with the existing holistic methods, all these probe images and gallery images were cropped to 128×128 pixels and properly aligned. Fig. 5.13 shows several cropped and aligned facial images from the AR dataset.

Table 5.7 records the recognition accuracy of various comparing approaches on the AR dataset, wherein S1-G and S1-S denote images of sunglasses and images of scarf from Session 1 respectively. Likewise, S2-G and S2-S denote images from S2. Note that some of the comparing algorithms have more than one gallery image per subject. The more gallery images per subject are used in experiment, the easier the recognition task would be. To compare with these methods, we also performed our RPSM with the same setting, *i.e.*, 7 gallery images per identity. We indicate the number of gallery images per subject used by various algorithms in the column of $\#g$. The recognition results show that RPSM achieve superior performance over the other state-of-the-art methods on S1-G and S2-G, and obtains satisfactory results on S1-S and S2-S. The good performance of our approach could be credited to our subset matching scheme, *i.e.*, the correspondence values of keypoints located among occlusion parts, such as sunglasses and scarf, are gradually set to zero during the matching process. In this way, outliers' impact on final distance metric is minimized. Only those matched keypoints in facial area are included into point set distance calculation. Another observation is that RPSM performs consistently better than MLERPM in all four scenarios, validating the effectiveness of adding constraint to affine transformation matrix.

To further showcase the effectiveness of our algorithm, we recognize the identity from the original whole probe images without crop and alignment. That is the size of probe image is 576×768 as shown in Fig. 5.13. In terms of gallery images, they were cropped to 128×128 . Hence, in this scenario, probe images were of different size from the gallery images and they were not aligned. The recognition results are reported in Table 5.8. Our RPSM performs the best throughout all four parts. One interesting phenomenon is that the recognition rates of MKD-SRC-GTP dropdrops a great deal a great deal. This is because it employs Harris-Laplacian detector [5] as interest point detector, which is more sensitive to corner than to blob. Hence, when the hair regions were included in the probe images, a large portion of interest points were

Table 5.7: Recognition accuracy (%) of the comparing algorithms on aligned AR dataset.

Method	# <i>g</i>	S1-G	S1-S	S2-G	S2-S
CRC [26]	7	68.50	90.50	N.A.	N.A.
RoBM [148]	7	84.50	80.70	N.A.	N.A.
ℓ_1 - ℓ_{struct} [19]	7	92.50	69.00	N.A.	N.A.
SLF-RKR [149]	7	100	100	91.30	96.00
GRRC_L1 [150]	7	90.70	95.30	50.30	87.30
SRC [20]	7	86.00	92.00	64.00	86.00
RPSM	7	100	100	92.00	95.33
Stringfaces [84]	1	88.00	96.00	76.00	88.00
CPD	1	79.33	81.67	51.67	66.67
MLERPM	1	82.00	85.33	59.67	71.33
LAIRPM	1	91.00	92.33	72.67	85.33
MKD-SRC-GTP	1	88.67	97.33	70	93.33
RPSM	1	93.00	94.33	76.67	86.00

Table 5.8: Recognition accuracy (%) of the comparing algorithms on original AR dataset.

Method	# <i>g</i>	S1-G	S1-S	S2-G	S2-S
LAIRPM	1	87.33	88.33	56.33	81.33
MLERPM	1	75.00	78.33	53.33	66.67
CPD	1	71.00	75.67	49.33	61.00
MKD-SRC-GTP	1	82.33	83.33	57.67	76.33
RPSM	1	88.67	90.33	63.67	85.67

detected among the hair region (hairs appear like corners to feature detector). However, hairs across sessions are not as stable as facial feature. Moreover, hair features are not discriminative.

Figure 5.14: Sample probe images in EYB dataset with random block occlusion. Their corresponding occlusion levels are listed underneath.

Table 5.9: Recognition accuracy (%) of various algorithms on EYB with different occlusion levels.

Occlusion	0%	10%	20%	30%	40%	50%
SRC [20]	100	100	99.80	98.50	90.30	65.30
GRRC.L1 [150]	100	100	100	100	96.50	87.40
SLF-RKR [149]	100	100	99.6	99.6	99.60	96.70
MKD-SRC	100	100	96.67	96.67	93.33	76.67
MLERPM	100	100	100	98.30	80.20	30.20
LAIRPM	100	100	100	96.67	66.67	53.33
RPSM	100	100	100	100	73.33	56.67

For the EYB dataset, we randomly chose 32 images of each subject as the gallery images, and the remaining 32 as probe. In our experiments, we synthesized contiguous-block-occluded images with occlusion levels ranging from 10% to 50%, by superimposing a correspondingly sized unrelated image randomly on each probe image. Some sample images are shown in Fig. 5.14. We compared our algorithm with SRC [20] and other three partial face recognition approaches, where we obtained some interesting results in Table 5.9. Before occlusion level arrives at 40%, our method performs better than or comparably with SRC and MKD-SRC-GTP, but it degrades drastically when the occlusion percent reaches more than 40%. While in the AR dataset, our method achieves quite satisfactory result where the percent of disguise for scarf is 40%. This is because in the experiment of AR dataset, disguise was either laid on the upper half or lower half of the face. Thus, discriminative features were almost half retained. While in this experiment, occlusion occurred randomly. For instance, in Fig. 5.14, when occlusion percent was 50%, the majority part of face area was occluded, making face match extremely difficult. Hence our method is suitable for scenarios where sufficient discriminative facial areas are available.

Figure 5.15: Sample example faces with various expressions from AR dataset. Facial images in the same column belong to the same identity. The images in the top row are samples from the gallery, and ones in the second and the third row are samples with smiling faces and screaming faces respectively.

5.3.5.5 Partial face recognition with facial expressions

We evaluated the robustness of our approach on partial face recognition with facial expressions on AR dataset. Specifically, for each person in AR, the neutral face in the first session was taken as the gallery images. Correspondingly, two smiling faces and two screaming faces from both sessions were chosen as the probe faces. These probe faces were further randomly transformed to simulate the scenario of the partial face recognition across facial expressions. Some sample faces are shown in Fig. 5.15.

Table 5.10 records the recognition rates of the comparing approaches, where “S1-Sm” denotes the smiling probe faces from Session 1 and “S2-Sc” the screaming faces from Session 2. It can be seen that recognizing identity from screaming faces is much more challenging than identifying smiling ones. Among the comparing approaches, our RPSM consistently outperforms the others.

5.3.6 RPSM vs. Holistic face recognition approach

While the proposed approach RPSM aims at recognizing partial faces, it’s of interest to compare it with an existing holistic face recognition approach with varying degrees of

Table 5.10: Recognition accuracy (%) of the comparing algorithms across facial expressions.

Method	S1-Sm	S2-Sm	S1-Sc	S2-Sc
LAIRPM	82.00	70.00	56.00	35.00
MLERPM	66.00	46.00	35.00	21.00
CPD	48.00	29.00	30.00	17.00
MKD-SRC-GTP	79.00	65.00	51.00	31.00
RPSM	88.00	74.00	60.00	48.00

Figure 5.16: Sample faces with various degrees of partialness from the LFW dataset. Facial images in the first column are the gallery images, and the ones from the second to the 7th column are the probe images with increasing degrees of partialness. The numbers s , θ , and t at the bottom of each column (except the first column) are the corresponding transformation parameters. The images in the top row belong to the same identity, and they form a positive match. The gallery face and the probe faces at the bottom row are from two different identities, and they constitute an impostor match.

“partialness”. In particular, we are interested to find out in which scenario our RPSM outperforms holistic approaches, and in which case holistic approaches are more favorable. We define the term of partialness as follows: the larger the facial area is cropped out, the greater partialness will be. To this end, we conducted face verification experiment on the LFW dataset. We used the Viola-Jones face detector to detect and crop out the facial regions of the View 2 set. The View 2 set consists of ten folds, each of which has 300 pairs of positive matches and 300 impostor matches. For each pair of image (either positive or impostor), we set the first image as the gallery and the second one as the probe. Furthermore, we transformed the probe facial images with varying

Figure 5.17: The ROC curves of RPSM and HDLBP working on 6 groups of LFW View 2 image sets.

rotation angles, scales and shifts. In particular, the rotation angle θ was gradually enlarged from 0° to 50° with 10° per step. The scaling parameter s was increased from 1 to 1.5. Similarly, the shifts on horizontal axis and vertical axis t rose from 0 to $0.25L$ with the incremental step being $0.05L$, where L was the width of the bounding box. In this case, we had probe images from 6 transformation groups, where the i th group had $s = 1 + 0.1i$, $\theta = i * 10^\circ$, $t = i * 0.05L$. Note that $i = 0, 1, \dots, 5$. During the transformation, the bounding box of the facial image remained unchanged. Some example faces are shown in Fig. 5.16. We can see that as the partialness increases, more facial areas are cropped out. Coupling these probe images with the gallery faces, we had 6 groups of View 2 sets, where the probe faces in the same group underwent the same geometric transformation. Hence, the larger the group number is, the more *partial* of the probe images will be. Specifically, the 0th group is a group of holistic probe facial images.

We compared the RPSM with the HDLBP approach as it is one of the best perform-

Table 5.11: Verification accuracy and corresponding standard deviations of RPSM and HDLBP on LFW.

Method	$\mu \pm S_E$
RPSM-G0	0.8314 ± 0.0014
HDLBP-G0	0.9413 ± 0.0029
RPSM-G1	0.8121 ± 0.0042
HDLBP-G1	0.8793 ± 0.0036
RPSM-G2	0.7924 ± 0.0114
HDLBP-G2	0.7583 ± 0.0052
RPSM-G3	0.7632 ± 0.0082
HDLBP-G3	0.6967 ± 0.0049
RPSM-G4	0.7411 ± 0.0053
HDLBP-G4	0.6117 ± 0.0047
RPSM-G5	0.7120 ± 0.0034
HDLBP-G5	0.5923 ± 0.0046

ing holistic face recognition approaches. The HDLBP was implemented as the same as the one in Section 5.3.5.1. Note that different from HDLBP, the RPSM doesn't involve any training processes. Fig. 5.17 shows the ROC curves of these two approaches on different groups of View 2 images. The notation is in the form of "Method-G#", *e.g.*, RPSM-G0 denotes the RPSM approach verifying the 0th group. HDLBP's ROC curves are delineated by solid lines, and RPSM by dash lines. Methods working on the same group have the same color. Table 5.11 shows the corresponding verification accuracy. It can be seen that RPSM achieves better recognition rate than HDLBP from Group 2 to Group 5. In particular, RPSM-G5 outperforms HDLBP-G3, and RPSM-G4 is comparable with HDLBP-G2. Moreover, the performance of HDLBP drops substantially from the group 2 onwards. This is because our HiCNN fails to work well when a facial component is missing. That's the reason why HDLBP performs miserably on the Group 3, 4, and 5. On the contrary, RPSM's performance decreases gradually as the partialness increases. There doesn't exist any significant performance degradation between two

consecutive groups. Therefore, for those partial faces where the current state-of-the-art face alignment methods fail, our RPSM is more favorable.

For the first two groups, RPSM is inferior to the HDLBP. This can be attributed to two reasons. First, RPSM is an unsupervised method which doesn't involve any training processes, while HDLBP is a supervised method. Second, HDLBP assumes that the face pairs have been properly aligned. Therefore, dense multi-scale local features can be extracted for comparison correspondingly. On the contrary, RPSM is designed for partial face recognition and it assumes the testing facial pairs are unaligned. Hence an alignment step is necessary. After alignment, the distance between two images is based on sparse matched features. In a nutshell, if the testing images are non-occluded holistic faces, the existing state-of-the-art holistic face recognition approaches are more competent than the proposed RPSM.

5.3.7 Discussions

Overall, our RPSM algorithms achieves satisfactory performance on partial face recognition and face verification, whether the probe faces are randomly transformed, whether they are aligned, whether they are occluded, whether they have exaggerated facial expressions or not. Specifically, we can make following observations from the above results:

- (i) RPSM obtains better performance than LAIRPM in all experiments. Note the mere difference between these two approaches lies on the matching schemes. In LAIRPM, it assumes that local structure is affine while the global structure is non-affine for the matching object. However, human face is non-rigid locally, and it is a rigid object except in the rare case of exaggerating facial expressions. On the contrary, our matching scheme deploys affine matching at first, which is to align two faces globally before detailed local warping (non-affine TPS transformation). Hence, LAIRPM's matching scheme is not as effective as ours in face recognition

scenarios.

- (ii) RPSM performs better than MLERPM in most cases. This is because RPSM adds a constraint on affine transformation matrix. This constraint prevents from abnormal face warping, while MLERPM is not free from this problem.
- (iii) RPSM consistently outperforms CPD in all experiments. CPD matches two feature sets merely based on geometric distribution, while RPSM considers both geometric consistency and textural similarity between feature sets.
- (iv) Regarding with MKD-SRC-GTP, our approach achieves better results except in face recognition on aligned AR dataset, where RPSM performs comparably with it (Table 5.7); and in face recognition with random block occlusion on the EYB dataset, where MKD-SRC-GTP has better performance when the occlusion rates reaches above 40% (Table 5.9). These two approaches differ with each other greatly on matching schemes. MKD-SRC-GTP piles local features together orderlessly and utilizes SRC to reconstruct probe feature set. It doesn't consider geometric information of local features. While in our approach, geometric features are taken into consideration directly by seeking transformation function between gallery and probe feature sets. That being said, SRC-based algorithms performs better when available discriminative features are scarce (Table 5.9). In this case, RPSM performs poorly as insufficient keypoints makes face alignment unreliable.
- (v) Compared with HDLBP, RPSM achieves favorable results when the current holistic face alignment methods fails to work well. In particular, the RPSM outperforms HDLBP significantly when a facial component is missing.

5.3.8 Computational Time

Our RPSM matching procedure is implemented in MATLAB with a noncommercial solver, `lpsolve` [151]. After the initial keypoint selection, our RPSM typically takes

around 10 ms to match a pair of probe and gallery keypoints on a desktop with core-i5 CPU @3.2GHZ, where the LP trust region shrinkage runs for 4-6 iterations. Note that the running time can be further shortened by implementing the method in C/C++.

5.4 Conclusion

In this chapter, we have proposed a partial face alignment and recognition method by using robust feature set matching. The proposed RPSM method is able to align the probe partial face to gallery facial images robustly even with the presence of occlusion, random partial crop, and exaggerated facial expressions. After face alignment, partial face recognition is achieved by measuring face similarity based on the proposed point set distance, which can be readily acquired with the face alignment result. The hallmark of the RPSM is its robust matching scheme, which considers both the geometric distribution consistency and the textural similarity. Moreover, constraint on the affine transformation is applied to prevent from unrealistic face warping. The proposed RPSM is particularly beneficial in the scenarios where current face alignment methods fail to perform well. This can be attributed to its capability to align partial faces, so long as there exist similar facial patches between the matching faces.

Experimental results on four widely used face datasets were presented to show the efficacy of RPSM on partial face alignment and recognition. For the partial face alignment, RPSM achieves much higher accuracy than HiCNN. In terms of the partial face recognition, the RPSM performs comparably with the state-of-the-art approaches in recognizing occluded faces, and it outperforms competing partial face recognition approaches on recognizing arbitrary facial patches. Moreover, we have conducted experiments to compare RPSM vs. HDLBP with increasing degrees of partialness. The results also validate the efficiency of RPSM in dealing with partial faces.

Figure 5.18: Eighteen face alignment examples by RPSM from AR dataset. Each row contains four pair of faces, which are two face alignment examples. Each example consists of two pairs of faces, where the left pair comprises the unaligned partial face and its corresponding holistic gallery image, whose local features chosen by the keypoint selection process are matched by green lines. Correspondingly, the face pair on the right shows the alignment result.

Chapter 6

Conclusions and Future Research

In this thesis we have proposed new approaches for face alignment and partial face recognition. In what follows, we summarize our research works and propose a few future research directions.

6.1 Conclusions

The face alignment and recognition are two challenging problems due to the large variations of facial appearances caused by changes of facial poses and illumination conditions, the presence of exaggerated facial expressions and occlusion, *etc.*. While the holistic face alignment approaches have achieved satisfactory results on near-frontal facial poses, their performance degrades substantially when facial images are largely rotated and the provided bounding boxes drifted. To address these challenges, we have proposed two new deep-neural-networks based face alignment approaches, which achieve superior alignment accuracy and robustness on three benchmarks. To deal with partial faces, we have further presented an approach which performs automatic partial face alignment and recognition in a single framework. The details are summarized as follows.

In Chapter 3, we have proposed a Cascaded Deep Auto-encoder Network (CDAN)

for holistic face alignment. The mainstream face alignment methods suffer from poor pose initialization and inaccurate pose update. To mitigate these two problems, we have designed two novel auto-encoder networks, namely GEDAN and LDAN. These two networks are the basic components that form the CDAN. Specifically, the GEDAN incorporates multiple representative exemplars into its top regression layer to improve the robustness of the pose initialization step against large pose variations. For LDAN, it applies a number of local auto-encoders (LAEs) to extract expressive pose-informative features and connects directly the output of these LAEs to the local regressors. In our experiment, we have verified the effectiveness of both the GEDAN and LDAN by comparing them to the Global-SAN and Local-SAN respectively. Moreover, experimental results have shown that CDAN consistently outperforms CFAN and performs comparably with other deep-learning based approaches. Furthermore, CDAN achieves real-time performance with Matlab on a common desktop (a desktop with core-i5 CPU @3.2GHZ) without customized design for speed up, *e.g.*, parallel computing.

In Chapter 4, we have devised a powerful face alignment approach named HiCNN to explicitly deal with the inaccurate face detection as well as large pose variations. In particular, the proposed HiCNN can detect facial landmarks from *arbitrarily* rotated facial images with inconsistent bounding boxes. The robustness and accuracy of HiCNN are brought by three specially tailored convolutional neural networks. Experimental results have shown that our HiCNN performs better than or comparably with the state-of-the-art face alignment methods in the case of ideal face detection, *i.e.*, the detected bounding box tightly encloses the facial area. Specifically, the HiCNN achieves 5.05 and 9.10 NRMSE on the common subset and the challenging subset respectively. Furthermore, HiCNN achieves much higher alignment accuracy than competing methods with drifted bounding boxes, indicating that HiCNN is more robust to inaccurate face detection. In addition, our approach achieves real-time performance with the speed of 50 FPS.

In Chapter 5, we have presented a partial face alignment and recognition method by

using a robust feature set matching method. Specifically, local feature sets of comparing facial images are matched iteratively by our RPSM approach. The outcomes of the matching are a point set correspondence matrix indicating matching keypoint pairs and a non-affine transformation function, the latter of which aligns the probe partial face to the gallery face automatically. Experimental results on four widely used face datasets were presented to show the efficiency of our approach in both the partial face alignment and recognition. For the partial face alignment, the RPSM performs much better than competing approaches, including local feature set matching methods, as well as state-of-the-art holistic face alignment methods. For the partial face recognition, the RPSM achieves superior recognition rates in identifying arbitrary face patches, and it performs comparably with SRC-based methods on recognizing occluded faces. Moreover, we have investigated the performance of RPSM against HDLBP with increasing degrees of “partialness”. The results show that RPSM is a promising complement to the existing state-of-the-art holistic face recognition methods, and the benefit of RPSM becomes more outstanding with the probe faces being more partial.

6.2 Recommendations for Future Research

Although the works reported in this thesis have achieved promising results, there exist a number of possible extensions to undertake in the future research.

- (i) For holistic face alignment, the current approaches are still incompetent in aligning profile faces. This is partly attributed to the lack of profile facial dataset with accurate landmark annotations. Even if the dataset is available, the landmark annotation formats of profile faces and the ones of frontal faces are likely to be different. It is desirable to devise a unifying face alignment approach to not only differentiate between profile and non-profile faces, but also detect facial landmarks with corresponding landmark annotation formats.

- (ii) For occluded faces, future research can be devoted to accurately detect facial landmark positions on severely occluded faces. If the majority of the facial area is occluded, the accuracy of our CDAN and HiCNN may degrade. Apart from facial landmark detection, it's desirable to detect the occlusion status of each landmark. Accurate occlusion status detection can greatly facilitate the subsequent face recognition stage, *e.g.*, we can apply larger weights on the non-occluded facial parts and decrease the weights corresponding to those occluded ones. A possible solution is to include the supervised information of occlusion statuses into the objective function Eq. (4.9). This approach, however, demands massive occluded facial images annotated with occlusion status labels.
- (iii) For partial face alignment and recognition on arbitrary facial patches, the proposed RPSM method is not robust enough. In particular, the performance of RPSM is subject to the repeatability of the local feature detector, which may be inconsistent suffering from large illumination changes. To address this problem, a more robust local feature detector is indispensable. Another shortcoming of RPSM is that it's unable to explicitly detect the occluded facial regions, *i.e.*, to delineate the occlusion boundary. Future research can be directed to devise a partial face alignment approach which is capable of detecting the visible facial regions from the background, the occluded facial parts (*e.g.*, facial parts occluded by sunglasses or scarf) and the facial components that are out of boundary.

Author's Publications

1. **Renliang Weng**, Jiwen Lu, and Yap-peng Tan, “Learning Cascaded Deep Auto-encoder Networks for Face Alignment”, submitted to IEEE Transactions on Information Forensics and Security, 2015.
2. **Renliang Weng**, Jiwen Lu, and Yap-peng Tan, “Hierarchical Convolutional Neural Networks for Robust Face Alignment”, submitted to IEEE Transactions on Image Processing, 2015.
3. **Renliang Weng**, Jiwen Lu, and Yap-peng Tan, “Robust Point Set Matching for Partial Face Recognition”, submitted to IEEE Transactions on Image Processing, 2014, major revision.
4. **Renliang Weng**, Jiwen Lu, Junlin Hu, Gao Yang, and Yap-peng Tan, “Robust feature set matching for partial face recognition”, in Proceedings of IEEE International Conference on Computer Vision (ICCV), 2013.
5. **Renliang Weng**, Jiwen Lu, Gao Yang, and Yap-peng Tan, “Multi-feature ordinal ranking for facial age estimation”, in Proceedings of IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), 2013.

Bibliography

- [1] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” University of Massachusetts, Amherst, Tech. Rep. 07-49, 2007.
- [2] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, “Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2233–2246, 2012.
- [3] “300 faces in-the-wild challenge,” <http://ibug.doc.ic.ac.uk/resources/300-W/>.
- [4] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *International Conference on Machine Learning*, 2010, pp. 807–814.
- [5] S. Liao, A. K. Jain, and S. Z. Li, “Partial face recognition: Alignment-free approach,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1193–1205, 2013.
- [6] X. Cao, Y. Wei, F. Wen, and J. Sun, “Face alignment by explicit shape regression,” *International Journal of Computer Vision*, vol. 107, no. 2, pp. 177–190, 2012.
- [7] X. Zhu and D. Ramanan, “Face detection, pose estimation, and landmark localization in the wild,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2879–2886.

- [8] X. Xiong and F. De la Torre, “Supervised descent method and its applications to face alignment,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 532–539.
- [9] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active appearance models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, 2001.
- [10] N. Wang, X. Gao, D. Tao, and X. Li, “Facial feature point detection: A comprehensive survey,” *arXiv preprint arXiv:1410.1037*, 2014.
- [11] G. B. Huang, V. Jain, and E. Learned-Miller, “Unsupervised joint alignment of complex images,” in *IEEE International Conference on Computer Vision*, 2007, pp. 1–8.
- [12] W. Deng, J. Hu, J. Lu, and J. Guo, “Transform-invariant pca: A unified approach to fully automatic face alignment, representation, and recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1275–1284, 2014.
- [13] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, 1997.
- [14] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, “Face recognition by independent component analysis,” *IEEE Transactions on Neural Networks*, vol. 13, no. 6, pp. 1450–1464, 2002.
- [15] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, “Face recognition using laplacianfaces,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, 2005.
- [16] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, “Graph embedding

- and extensions: a general framework for dimensionality reduction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, 2007.
- [17] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 1991, pp. 586–591.
- [18] E. Elhamifar and R. Vidal, “Robust classification using structured sparse representation,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1873–1879.
- [19] K. Jia, T.-H. Chan, and Y. Ma, “Robust and practical face recognition via structured sparsity,” in *European Conference on Computer Vision*, 2012, pp. 331–344.
- [20] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, 2009.
- [21] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.
- [22] Y. Sun, X. Wang, and X. Tang, “Deep learning face representation from predicting 10,000 classes,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1891–1898.
- [23] Z. Zhu, P. Luo, X. Wang, and X. Tang, “Multi-view perceptron: a deep model for learning face identity and view representations,” in *Advances in Neural Information Processing Systems*, 2014, pp. 217–225.
- [24] C. Lu and X. Tang, “Surpassing human-level face verification performance on lfw with gaussianface,” *arXiv preprint arXiv:1404.3840*, 2014.

- [25] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” *arXiv preprint arXiv:1503.03832*, 2015.
- [26] L. Zhang, M. Yang, and X. Feng, “Sparse representation or collaborative representation: Which helps face recognition?” in *IEEE International Conference on Computer Vision*, 2011, pp. 471–478.
- [27] Y. Liu, F. Wu, Z. Zhang, Y. Zhuang, and S. Yan, “Sparse representation using nonnegative curds and whey,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3578–3585.
- [28] I. Matthews and S. Baker, “Active appearance models revisited,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004.
- [29] G. Tzimiropoulos and M. Pantic, “Optimization problems for fast aam fitting in-the-wild,” in *IEEE International Conference on Computer Vision*, 2013, pp. 593–600.
- [30] D. Cristinacce and T. F. Cootes, “Feature detection and tracking with constrained local models,” in *British Machine Vision Conference*, pp. 929–938.
- [31] J. M. Saragih, S. Lucey, and J. F. Cohn, “Deformable model fitting by regularized landmark mean-shift,” *International Journal of Computer Vision*, vol. 91, no. 2, pp. 200–215, 2011.
- [32] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, “Robust discriminative response map fitting with constrained local models,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3444–3451.
- [33] X. P. Burgos-Artizzu, P. Perona, and P. Dollár, “Robust face landmark estimation under occlusion,” in *IEEE International Conference on Computer Vision*, 2013, pp. 1513–1520.

- [34] J. Zhang, S. Shan, M. Kan, and X. Chen, “Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment,” in *European Conference on Computer Vision*, 2014, pp. 1–16.
- [35] Y. Sun, X. Wang, and X. Tang, “Deep convolutional network cascade for facial point detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3476–3483.
- [36] V. Kazemi and J. Sullivan, “One millisecond face alignment with an ensemble of regression trees,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1867–1874.
- [37] S. Ren, X. Cao, Y. Wei, and J. Sun, “Face alignment at 3000 FPS via regressing local binary features,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1685–1692.
- [38] S. Shan, Y. Chang, W. Gao, B. Cao, and P. Yang, “Curse of mis-alignment in face recognition: problem and a novel mis-alignment learning solution,” in *IEEE International Conference on Automatic Face and Gesture Recognition*, 2004, pp. 314–320.
- [39] P. N. Belhumeur, D. W. Jacobs, D. Kriegman, and N. Kumar, “Localizing parts of faces using a consensus of exemplars,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 545–552.
- [40] F. Zhou, J. Brandt, and Z. Lin, “Exemplar-based graph matching for robust facial landmark localization,” in *IEEE International Conference on Computer Vision*, 2013, pp. 1025–1032.
- [41] J. Xing, Z. Niu, J. Huang, W. Hu, and S. Yan, “Towards multi-view and partially-occluded face alignment,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1829–1836.

- [42] S. Zheng, P. Sturges, and P. Torr, "Approximate structured output learning for constrained local models with application to real-time facial feature detection and tracking on low-power devices," in *IEEE International Conference on Automatic Face and Gesture Recognition*, 2013, pp. 1–8.
- [43] G. Tzimiropoulos and M. Pantic, "Gauss-newton deformable part models for face alignment in-the-wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1851–1858.
- [44] P. D. Sozou, T. F. Cootes, C. J. Taylor, E. Di Mauro, and A. Lanitis, "Non-linear point distribution modelling using a multi-layer perceptron," *Image Vision Comput.*, vol. 15, no. 6, pp. 457–463, 1997.
- [45] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool, "Real-time facial feature detection using conditional regression forests," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2578–2585.
- [46] S. Zhu, C. Li, C. C. Loy, and X. Tang, "Face alignment by coarse-to-fine shape searching," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4998–5006.
- [47] Z.-H. Feng, P. Huber, J. Kittler, W. Christmas, and X.-J. Wu, "Random cascaded-regression copse for robust facial landmark detection," *SPL*, vol. 22, no. 1, pp. 76–80, 2015.
- [48] P. Luo, X. Wang, and X. Tang, "Hierarchical face parsing via deep learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2480–2487.
- [49] Y. Wu, Z. Wang, and Q. Ji, "Facial feature tracking under varying facial expressions and face poses based on restricted boltzmann machines," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3452–3459.

- [50] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, “Learning and transferring multi-task deep representation for face alignment,” *arXiv preprint arXiv:1408.3967*, 2014.
- [51] G. Ghiasi and C. C. Fowlkes, “Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1899–1906.
- [52] X. Yu, Z. Lin, J. Brandt, and D. N. Metaxas, “Consensus of regression for occlusion-robust facial feature localization,” in *European Conference on Computer Vision*, 2014, pp. 105–118.
- [53] X. Gao, Y. Su, X. Li, and D. Tao, “A review of active appearance models,” *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 40, no. 2, pp. 145–158, 2010.
- [54] X. W. Hou, S. Z. Li, H. Zhang, and Q. Cheng, “Direct appearance models,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. I–828.
- [55] A. U. Batur and M. H. Hayes, “Adaptive active appearance models,” *IEEE Transactions on Image Processing*, vol. 14, no. 11, pp. 1707–1721, 2005.
- [56] X. Liu, “Generic face alignment using boosted appearance model,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
- [57] J. Saragih and R. Göcke, “Learning aam fitting through simulation,” *Pattern Recognition*, vol. 42, no. 11, pp. 2628–2636, 2009.
- [58] G. Papandreou and P. Maragos, “Adaptive and constrained algorithms for inverse compositional active appearance model fitting,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

- [59] F. Tang and B. Deng, “Facial expression recognition using aam and local facial features,” in *International Conference on Natural Computation*, vol. 2, 2007, pp. 632–635.
- [60] E. Antonakos and S. Zafeiriou, “Automatic construction of deformable models in-the-wild,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1813–1820.
- [61] L. Liang, F. Wen, Y.-Q. Xu, X. Tang, and H.-Y. Shum, “Accurate face alignment using shape constrained markov network,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, pp. 1313–1319.
- [62] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, “Incremental face alignment in the wild,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1859–1866.
- [63] T. Baltrušaitis, P. Robinson, and L.-P. Morency, “Continuous conditional neural fields for structured regression,” in *European Conference on Computer Vision*, 2014, pp. 593–608.
- [64] X. Zhao, T.-K. Kim, and W. Luo, “Unified face analysis by iterative multi-output random forests,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1765–1772.
- [65] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun, “Joint cascade face detection and alignment,” in *European Conference on Computer Vision*. Springer, 2014, pp. 109–122.
- [66] Z. Zhang, P. Luo, C. Loy, and X. Tang, “Facial landmark detection by deep multi-task learning,” in *European Conference on Computer Vision*, 2014, pp. 94–108.

- [67] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, “Extensive facial landmark localization with coarse-to-fine convolutional network cascade,” in *IEEE International Conference on Computer Vision Workshops*, 2013, pp. 386–391.
- [68] E. G. Learned-Miller, “Data driven image models through continuous joint alignment,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 2, pp. 236–250, 2006.
- [69] M. Cox, S. Sridharan, S. Lucey, and J. Cohn, “Least squares congealing for unsupervised alignment of images,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
- [70] B. J. Frey and N. Jojic, “Transformed component analysis: Joint estimation of spatial transformations and image components,” in *IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1190–1196.
- [71] H. Schweitzer, “Optimal eigenfeature selection by optimal image registration,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 1999.
- [72] A. Vedaldi, G. Guidi, and S. Soatto, “Joint data alignment up to (lossy) transformations,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
- [73] W. Deng, J. Hu, L. Liu, and J. Guo, “Transformed principal gradient orientation for robust and precise batch face alignment,” in *Asian Conference on Computer Vision*, 2014, pp. 708–723.
- [74] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, “Attribute and simile classifiers for face verification,” in *IEEE International Conference on Computer Vision*, 2009, pp. 365–372.
- [75] X. Tan, S. Chen, Z.-H. Zhou, and F. Zhang, “Face recognition from a single image per person: A survey,” *Pattern recognition*, vol. 39, no. 9, pp. 1725–1745, 2006.

- [76] X. Zhang and Y. Gao, "Face recognition across pose: A review," *Pattern Recognition*, vol. 42, no. 11, pp. 2876–2896, 2009.
- [77] V. Blanz, S. Romdhani, and T. Vetter, "Face identification across different poses and illuminations with a 3d morphable model," in *IEEE International Conference on Automatic Face and Gesture Recognition*, 2002, pp. 192–197.
- [78] C. Liu, "Capitalize on dimensionality increasing techniques for improving face recognition grand challenge performance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 725–737, 2006.
- [79] N. Ramanathan and R. Chellappa, "Face verification across age progression," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3349–3361, 2006.
- [80] Y. Su, S. Shan, X. Chen, and W. Gao, "Hierarchical ensemble of global and local classifiers for face recognition," *IEEE Transactions on Image Processing*, vol. 18, no. 8, pp. 1885–1896, 2009.
- [81] K. Delac, M. Grgic, and S. Grgic, "Independent comparative study of pca, ica, and lda on the feret data set," *International Journal of Imaging Systems and Technology*, vol. 15, no. 5, p. 252, 2005.
- [82] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *Acm Computing Surveys (CSUR)*, vol. 35, no. 4, pp. 399–458, 2003.
- [83] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [84] W. Chen and Y. Gao, "Recognizing partially occluded faces from a single sample per class using string-based matching," in *European Conference on Computer Vision*, 2010, pp. 496–509.

- [85] T. Ahonen, A. Hadid, and M. Pietikainen, “Face description with local binary patterns: Application to face recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, 2006.
- [86] R. Min, A. Hadid, and J. Dugelay, “Improving the recognition of faces occluded by facial accessories,” in *IEEE International Conference on Automatic Face and Gesture Recognition*, 2011, pp. 442–447.
- [87] K. Pan, S. Liao, Z. Zhang, S. Z. Li, and P. Zhang, “Part-based face recognition using near infrared images,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–6.
- [88] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang, “Probabilistic elastic matching for pose variant face verification,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3499–3506.
- [89] A. Li, S. Shan, X. Chen, and W. Gao, “Face recognition based on non-corresponding region matching,” in *IEEE International Conference on Computer Vision*, 2011, pp. 1060–1067.
- [90] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Fisher vector faces in the wild,” in *British Machine Vision Conference*, 2013.
- [91] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli, “On the use of SIFT features for face authentication,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2006, pp. 35–35.
- [92] C. Geng and X. Jiang, “Face recognition based on the multi-scale local image structures,” *Pattern Recognition*, vol. 44, no. 10, pp. 2565–2575, 2011.
- [93] P. Dreuw, P. Steingrube, H. Hanselmann, H. Ney, and G. Aachen, “SURF-face:

- Face recognition under viewpoint consistency constraints,” in *British Machine Vision Conference*, 2009, pp. 1–11.
- [94] X. Tan and B. Triggs, “Enhanced local texture feature sets for face recognition under difficult lighting conditions,” *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1635–1650, 2010.
- [95] D. Chen, X. Cao, F. Wen, and J. Sun, “Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3025–3032.
- [96] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, “Bayesian face revisited: A joint formulation,” in *European Conference on Computer Vision*, 2012, pp. 566–579.
- [97] X. Cai, C. Wang, B. Xiao, X. Chen, and J. Zhou, “Deep nonlinear metric learning with independent subspace analysis for face verification,” in *20th ACM International Conference on Multimedia*, 2012, pp. 749–752.
- [98] J. Hu, J. Lu, and Y.-P. Tan, “Discriminative deep metric learning for face verification in the wild,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1875–1882.
- [99] Z. Cui, W. Li, D. Xu, S. Shan, and X. Chen, “Fusing robust face region descriptors via multiple metric learning for face recognition in the wild,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3554–3561.
- [100] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, “Information-theoretic metric learning,” in *International Conference on Machine learning*. ACM, 2007, pp. 209–216.
- [101] M. Guillaumin, J. Verbeek, and C. Schmid, “Is that you? metric learning ap-

- proaches for face identification,” in *International Conference on Computer Vision*, 2009, pp. 498–505.
- [102] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, “Large scale metric learning from equivalence constraints,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2288–2295.
- [103] K. Q. Weinberger, J. Blitzer, and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” in *Advances in Neural Information Processing Systems*, 2005, pp. 1473–1480.
- [104] ———, “Distance metric learning for large margin nearest neighbor classification,” in *Advances in Neural Information Processing Systems*, 2006, p. 1473.
- [105] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. Salakhutdinov, “Neighbourhood components analysis,” in *Advances in neural information processing systems*, 2004, pp. 513–520.
- [106] R. Weng, J. Lu, J. Hu, G. Yang, and Y.-P. Tan, “Robust feature set matching for partial face recognition,” in *IEEE International Conference on Computer Vision*, 2013, pp. 601–608.
- [107] D. Conte, P. Foggia, C. Sansone, and M. Vento, “Thirty years of graph matching in pattern recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 03, pp. 265–298, 2004.
- [108] L. Torresani, V. Kolmogorov, and C. Rother, “Feature correspondence via graph matching: Models and global optimization,” in *European Conference on Computer Vision*, 2008, pp. 596–609.
- [109] H. Chui and A. Rangarajan, “A new point matching algorithm for non-rigid regis-

- tration,” *Computer Vision and Image Understanding*, vol. 89, no. 2, pp. 114–141, 2003.
- [110] K. Grauman and T. Darrell, “The pyramid match kernel: Discriminative classification with sets of image features,” in *IEEE International Conference on Computer Vision*, vol. 2, 2005, pp. 1458–1465.
- [111] B. Jian and B. C. Vemuri, “Robust point set registration using gaussian mixture models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1633–1645, 2011.
- [112] L. Maier-Hein, A. M. Franz, T. R. dos Santos, M. Schmidt, M. Fangerau, H. Meinzer, and J. M. Fitzpatrick, “Convergent iterative closest-point algorithm to accomodate anisotropic and inhomogenous localization error,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 8, pp. 1520–1532, 2012.
- [113] F. Escolano, E. Hancock, and M. Lozano, “Graph matching through entropic manifold alignment,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2417–2424.
- [114] J. Ma, J. Zhao, J. Tian, Z. Tu, and A. L. Yuille, “Robust estimation of nonrigid transformation for point set registration,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2147–2154.
- [115] F. Zhou and F. De la Torre, “Factorized graph matching,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 127–134.
- [116] J. R. Ullmann, “An algorithm for subgraph isomorphism,” *Journal of the ACM (JACM)*, vol. 23, no. 1, pp. 31–42, 1976.
- [117] M. Pelillo, “Replicator equations, maximal cliques, and graph isomorphism,” *Neural Computation*, vol. 11, no. 8, pp. 1933–1955, 1999.

- [118] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, “A (sub) graph isomorphism algorithm for matching large graphs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 10, pp. 1367–1372, 2004.
- [119] S. Umeyama, “An eigendecomposition approach to weighted graph matching problems,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 5, pp. 695–703, 1988.
- [120] G. L. Scott and H. C. Longuet-Higgins, “An algorithm for associating the features of two images,” *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 244, no. 1309, pp. 21–26, 1991.
- [121] T. Caelli and S. Kosinov, “An eigenspace projection clustering method for inexact graph matching,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 4, pp. 515–519, 2004.
- [122] M. Leordeanu and M. Hebert, “A spectral technique for correspondence problems using pairwise constraints,” in *IEEE International Conference on Computer Vision*, vol. 2, 2005, pp. 1482–1489.
- [123] H. Almohamad and S. O. Duffuaa, “A linear programming approach for the weighted graph matching problem,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 5, pp. 522–525, 1993.
- [124] H. Bunke, “Graph matching: Theoretical foundations, algorithms, and applications,” in *Proc. Vision Interface*, vol. 2000, 2000, pp. 82–88.
- [125] J. Pokrass, A. M. Bronstein, M. M. Bronstein, P. Sprechmann, and G. Sapiro, “Sparse modeling of intrinsic correspondences,” in *Computer Graphics Forum*, vol. 32, no. 2pt4, 2013, pp. 459–468.

- [126] H. Li, X. Huang, and L. He, “Object matching using a locally affine invariant and linear programming techniques,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 411–424, 2013.
- [127] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. von der Malsburg, “Face recognition by elastic bunch graph matching,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 775–779, 1997.
- [128] Q. Liao, J. Z. Leibo, Y. Mroueh, and T. A. Poggio, “Can a biologically-plausible hierarchy effectively replace face detection, alignment, and recognition pipelines?” *CoRR*, vol. abs/1311.4082, 2013.
- [129] Y. Chauvin and D. E. Rumelhart, *Backpropagation: Theory, Architectures, and Applications*. Psychology Press, 1995.
- [130] P. Dollár, P. Welinder, and P. Perona, “Cascaded pose regression,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1078–1085.
- [131] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.
- [132] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, “300 faces in-the-wild challenge: The first facial landmark localization challenge,” in *IEEE International Conference on Computer Vision Workshops*, 2013, pp. 397–403.
- [133] G. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [134] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, “Interactive facial feature localization,” in *European Conference on Computer Vision*, 2012, pp. 679–692.

- [135] A. Vedaldi and K. Lenc, “Matconvnet – convolutional neural networks for matlab,” *CoRR*, vol. abs/1412.4564, 2014.
- [136] S. Zhu, C. Li, C. C. Loy, and X. Tang, “Transferring landmark annotations for cross-dataset face alignment,” *CoRR*, vol. abs/1409.0602, 2014.
- [137] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” University of Massachusetts, Amherst, Tech. Rep. 07-49, 2007.
- [138] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [139] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *European Conference on Computer Vision*, 2006, pp. 404–417.
- [140] L. Juan and O. Gwun, “A comparison of sift, pca-sift and surf,” *International Journal of Image Processing*, vol. 3, no. 4, pp. 143–152, 2009.
- [141] Z. Li, G. Liu, Y. Yang, and J. You, “Scale-and rotation-invariant local binary pattern using scale-adaptive texton and subuniform-based circular shift,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2130–2140, 2012.
- [142] F. L. Bookstein, “Principal warps: Thin-plate splines and the decomposition of deformations,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 6, pp. 567–585, 1989.
- [143] H. Jiang, M. S. Drew, and Z.-N. Li, “Matching by linear programming and successive convexification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 959–975, 2007.
- [144] M. Martinez and R. Benavente, “The AR face database,” 1998.

- [145] A. Myronenko and X. Song, “Point set registration: Coherent point drift,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2262–2275, 2010.
- [146] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, “Comparing images using the hausdorff distance,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, 1993.
- [147] Y. Rubner, C. Tomasi, and L. J. Guibas, “The earth mover’s distance as a metric for image retrieval,” *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.
- [148] Y. Tang, R. Salakhutdinov, and G. Hinton, “Robust boltzmann machines for recognition and denoising,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2264–2271.
- [149] M. Yang, L. Zhang, S. C.-K. Shiu, and D. Zhang, “Robust kernel representation with statistical local features for face recognition,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 6, pp. 900–912, 2013.
- [150] M. Yang, L. Zhang, S. C. Shiu, and D. Zhang, “Gabor feature based robust representation and classification for face recognition with gabor occlusion dictionary,” *Pattern Recognition*, vol. 46, no. 7, pp. 1865–1878, 2013.
- [151] lpsolve:<http://sourceforge.net/projects/lpsolve/>, 2012.