ABSTRACT

Title of dissertation: DOMAIN ADAPTATION FOR UNCONSTRAINED FACE VERIFICATION AND IDENTIFICATION

Boyu Lu, Doctor of Philosophy, 2019

Dissertation directed by: Professor Rama Chellappa
Department of Electrical and
Computer Engineering

Face recognition has been receiving consistent attention in computer vision community for over three decades. Although recent advances in deep convolutional neural networks (DCNNs) have pushed face recognition algorithms to surpass human performance in most controlled situations, the unconstrained face recognition performance is still far from satisfactory. This is mainly because the domain shift between training and test data is substantial when faces are captured under extreme pose, blur or other covariates variations. In this dissertation, we study the effects of covariates and present approaches of mitigating the domain mismatch to improve the performance of unconstrained face verification and identification.

To study how covariates affect the performance of deep neural networks on the large-scale unconstrained face verification problem, we implement five state-of-the-art deep convolutional networks (DCNNs) and evaluate them on three challenging covariates datasets. In total, seven covariates are considered: pose (yaw and roll), age, facial hair, gender, indoor/outdoor, occlusion (nose and mouth visibility, and

forehead visibility), and skin tone. Some of the results confirm and extend the findings of previous studies, while others are new findings that were rarely mentioned before or did not show consistent trends. In addition, we demonstrate that with the assistance of gender information, the quality of a pre-curated noisy large-scale face dataset can be further improved.

Based on the results of this study, we propose four domain adaptation methods to alleviate the effects of covariates. First, since we find that pose is a key factor for performance degradation, we propose a metric learning method to alleviate the effects of pose on face verification performance. We learn a joint model for face and pose verification tasks and explicitly discourage information sharing between the identity and pose metrics. Specifically, we enforce an orthogonal regularization constraint on the learned projection matrices for the two tasks leading to making the identity metrics for face verification more pose-robust. Extensive experiments are conducted on three challenging unconstrained face datasets that show promising results compared to state-of-the-art methods.

Second, to tackle the negative effects brought by image blur, we propose two approaches. The first approach is an incremental dictionary learning method to mitigate the distribution difference between sharp training data and blurred test data. Some blurred faces called supportive samples are selected, which are used for building more discriminative classification models and act as a bridge to connect the two domains. Second, we propose an unsupervised face deblurring approach based on disentangled representations. The disentanglement is achieved by splitting the content and blur features in a blurred image using content encoders and blur

encoders. An adversarial loss is added on deblurred results to generate visually realistic faces. We conduct extensive experiments on two challenging face datasets that show promising results.

Finally, apart from the effects of pose and blur, face verification performance also suffers from the generic domain mismatch between source and target faces. To tackle this problem, we propose a template adaptation method for template-based face verification. A template-specific metric is trained to adaptively learn the discriminative information between test templates and the negative training set, which contains subjects that are mutually exclusive to subjects in test templates. Extensive experiments on two challenging face verification datasets yield promising results compared to other competitive methods.

DOMAIN ADAPTION FOR UNCONSTRAINED FACE VERIFICATION AND IDENTIFICATION

by

Boyu Lu

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2019

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor Joseph F. JaJa
Professor Behtash Babadi
Dr. Carlos Castillo
Professor Ramani Duraiswami, Dean's Representative

Dedication

To my father, for supporting and pushing me all the way to become a Ph.D.

Acknowledgments

Seven years ago, when I started this journey to pursue the Ph.D. degree, I never thought I would experience so much, up and downs, joys and disappointments, uncertainties and doubts. As I finally reach this stage, I owe my gratitude to all the people who supported and helped me and without whom this dissertation would not be possible.

First and foremost, I would like to thank my advisor, Professor Rama Chellappa, for his constant support, encouragement and inspiration. I was given the maximum possible freedom to explore the research topics I was interested in. Whenever I felt frustrated, his unique sense of humor always kept me up; whenever I made some research progress, his simple but powerful encouragement motivated me to do better. He sets a perfect model for being a successful and admired professor: works extremely hard, always humble and patient, helps others without any reservation. I am so lucky to be advised by him and I will never forget his words:"being a Ph.D. student is privilege and you should value this opportunity."

I would also like to thank Professor JaJa, Professor Babadi, Professor Duraiswami and Dr. Castillo for kindly agreeing to serve on my advisory committee and providing valuable feedbacks and suggestions to make this dissertation better.

I am thankful to all my co-authors and collaborators Dr. Nasser Nasrabadi, Dr. Jun-Cheng Chen, and Jingxiao Zheng, with a special mention to Dr. Jun-Cheng Chen. Dr. Chen and I worked together in almost every paper in this dissertation and I benefited so much from his creative ideas and seriousness.

In addition, my graduate life has been greatly enriched by my fellow colleagues in Rama's group, especially Dr. Maya Kabkab, Dr. Ashish Srivastava, Dr. Jie Ni, Dr. Jingjing Zheng, Dr. Ching-Hui Chen, Dr. Emily Hand, Dr. Rajeev Ranjan, Dr. Swami Sankaranarayanan, Hui Ding, Hongyu Xu, and Pengcheng Xu.

I am grateful for the administrative help I have received from Ms. Melanie Prange, Ms. Vivian Lu, Mr. Bill Churma, Ms. Arlene Schenk, and Ms. Janice Perone. They are always pateient and willing to help me go through many annoying administrative procedures.

I would also like to express my deepest gratitude to my parents, my grandma, and my aunt, who stand by me and give me inexhaustible love. In particular, I want to thank my wife, Liuwei Zhao. Her encouragement and companionship helped me go through many difficult moments. Without her, I will not be who I am.

Finally, my research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2014-14071600012. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Motivation

Face recognition has been an active research area in computer vision community for decades. In general, face recognition can be divided into two sub-problems: face verification and face identification. The task of face verification is to verify whether a pair of face images/templates belong to the same subject. In contrast, face identification aims to match a query face images/template to one of the enrolled gallery subjects or to classify it as an unseen subject. Recently, due to the rapid development of deep convolutional neural networks (DCNNs), face recognition performance has improved significantly and state-of-the-art face recognition algorithms have surpassed human performance [1–6].

Despite the promising performance of DCNNs, some recent works have observed that unconstrained face recognition performance is still significantly affected by many covariates [7–10]. Therefore, the problem of unconstrained face recognition under extreme pose, illumination, blur and other covariates variations remains unsolved. The main challenges of unconstrained face recognition come from two aspects. First, since face images of subjects are captured in a non-cooperative way, the pose of the face and body may vary significantly. Second, images usually are

not taken by professional photographers and often suffer from blur, occlusion, and low resolution. In this dissertation, we focus on designing more robust models to tackle these challenges.

1.2 Overview

Although there have been many previous works that have studied the effects of covariates on the face verification performance, most of them are outdated — most studies were conducted before the emergence of deep networks and the evaluation datasets were small and constrained. Therefore, in the first part of this dissertation, we perform comprehensive experiments to investigate the effects of these covariates on the performance of the state-of-the-art deep face models.

From the experimental results, we find that two factors significantly impair the performance: pose and blur. To mitigate the negative effects brought by pose, in Chapter 3, we propose a pose robust metric learning approach to explicitly suppress the pose information contained in deep features. For blur effects, we present two methods in Chapter 4 and 5 to tackle this problem in two different ways. The first method is based on incremental dictionary learning. It reduces the domain distance between sharp and blurred faces in feature space. In contrast, the second approach directly restore the blurred faces, which reduce the domain mismatch in pixel space. In addition, we also propose a generic method for template-based face verification in Chapter 6.

In the following sections of this chapter, we introduce more details of our study

on covariates and the proposed methods for reducing their effects.

1.3 Covariates Effects on Unconstrained Face Verification

Covariates are factors that usually have an undesirable influence on face verification performance (*e.g.*, gender induces different human facial appearance characteristics in nature.). Some covariates represent different aspects of faces such as pose, expression and age, some covariates represent subject-specific intrinsic characteristics like gender, race and skin tone, and other covariates reflect extrinsic factors of images, such as illuminations, occlusion and resolution. Analyzing the effects of these covariates can not only help understand fundamental challenges in face verification, but also provide insights for improving existing face verification algorithms.

In Chapter 2, we investigate two important problems: a) how different covariates affect the performance of state-of-the-art DCNNs for unconstrained face verification; b) how to utilize gender information to improve face verification performance. For the first problem, we implement five state-of-the-art face DCNNs and evaluate them on three challenging covariate protocols. By conducting extensive experiments on these datasets, we observe many interesting behaviors for different covariates. Some of our findings support conclusions drawn from previous studies. For example, extreme yaw angles do substantially degrade the performance [11] and outdoor images are harder to be recognized [12]. Meanwhile, we also find some results which extend the findings of previous works due to the availability of larger datasets. For example, most previous studies show that face recognition algorithms

usually achieve better performance on older subjects than younger subjects [13, 14]. But in their studies, most of the enrolled subjects are under 40 years old. However, our experiments with significantly more subjects with a wider age range show that the performance does not monotonically increase as age progresses. The performance increases from age group $[0, 19]$ to age group $[35, 49]$ but begins to drop for age group $[50, 65]$ and 65+. The results demonstrate that neither too young nor too old people are easy to recognize, but the recognition results for very young people (*i.e.*, $[0, 19]$) are the worst. Moreover, we are able to better evaluate some covariates like gender where previous works reached contradictory conclusions [13]. Our experiments show that in general, males are easier to match than females. However, when we combine gender with other covariates (age, skin tone) to investigate their mixed effects, we find that the face verification performance for females becomes better than males' for older age group and darker skin tones. Finally, some of our results are surprising yet rarely analyzed in previous papers. One example is that roll variations greatly affect verification performance in unconstrained situation. Since most previous studies may have used manually aligned faces, roll variation was not a significant factor in their studies. However, in unconstrained environments, face alignment becomes a key component and our finding shows that performance variations might result from the fact that face alignment algorithms fail to work perfectly for faces in extreme roll angles.

For the second problem, we utilize gender information to curate a noisy large-scale face dataset. Specifically, we find that the curated MS-Celeb1M [15, 16] still contains many noisy labels where some subjects still contain images from differ-

ent genders. Training using the noisy data may potentially hurt the discriminative capability of deep models and degrade their performance, especially in low FAR regions (10^{-5}, 10^{-6}, etc). Therefore, we leverage gender information to further curate the training set and remove subjects mixed with images of both males and females. After retraining the model using the curated data, the performance improves at low FARs.

1.4 Pose-Robust Face Verification by Exploiting Competing Tasks

Among the face covariates, pose variation is one of the most difficult challenges as it has great impact on face recognition performance even when the best DCNNs algorithms are used. This has led to growing interest in pose-robust face recognition in recent years [17]. Li *et al.* [18] designed a pose-invariant representation for faces by extracting densely sampled local features and training a Gaussian mixture model (GMM) on them. The GMM captures the spatial-appearance distribution of face images by augmenting local features with their locations. Zhu *et al.* [19] proposed a two-stage deep neural network to frontalize the off-frontal face images. The first module was used for feature extraction while the second module reconstructed the faces at a canonical view. Kan *et al.* [20] learned a discriminant common space for faces from different poses by maximizing the between-class variations and minimizing the within-class variations. Ding *et al.* [21] generated a generic 3D model and transformed the profile faces to synthesized partial frontal faces. Then patch-based face representations were used for face matching. AbdAlmageed *et al.* [22]

utilized generic 3D models to generate synthetic faces in different poses and used pose-specific CNNs to extract features. The similarity between two face images was computed as fusion of the pose-specific feature similarities.

Different from the works mentioned above, in Chapter 3, we tackle this problem by learning pose-robust metrics in which pose-sensitive information is explicitly mitigated. To achieve this goal, we introduce an auxiliary task called pose verification (*i.e.,* checking whether the two faces are in the same pose.) and exploit the competitive relationships between the auxiliary task (pose verification) and the main task of face verification. More specifically, we propose a multi-task framework where the face verification and pose verification models are learned simultaneously. Based on the intuition that the metrics for the two tasks are competing with each other, we jointly learn the projection matrices for the two tasks and add an orthogonal regularization constraint. The learned metric for face verification is thus robust to pose variations and overcomes the pose mismatch between training and test data to some extent. Experimental results on three challenging face datasets demonstrate promising performances as compared to other competing methods.

1.5 Blurred Face Recognition by Incremental Dictionary Learning-based Domain Adaptation and Unsupervised Face Deblurring

Image blurring is another important factor that adversely affects the quality of images and thus significantly degrades the performance of face recognition algorithms [23]. To address this problem, two types of methods have been considered.

The first type of methods utilizes the idea of domain adaptation which explicitly reduces the domain dissimilarity while the second category of approaches directly applies blind image deblurring algorithms to restore the latent sharp image from a blurred image.

Domain adaption methods originate from the observation that training and test data are often drawn from different latent distributions for many real applications. For instance, classifiers which are trained on samples in frontal or near-frontal poses may be called upon to recognize non-frontal poses; face verification metrics which are learned from pairs with similar resolutions and illuminations may be used for verifying pairs with very different acquisition conditions. This domain mismatch violates the key assumption of the traditional supervised learning methods and therefore leads to significant performance drop.

In Chapter 4, we propose an incremental dictionary learning method where some target data called supportive samples are selected to assist adaptation. Supportive samples are close to the source domain and have two properties: first, their predicted class labels are reliable and can be used for building more discriminative classification models; second, they act as a bridge to connect the two domains and reduce the domain mismatch. Theoretical analysis shows that both properties are important for adaptation, supporting the idea of adding supportive samples to the source domain. A stopping criterion is designed to guarantee that the domain mismatch decreases monotonically during adaptation. Experimental results on blurred face datasets and object classification tasks show that the proposed approach performs better than many state-of-the-art methods.

Blind image deblurring aims to directly reconstruct sharp images. Most conventional methods formulate the image deblurring task as a blur kernel estimation problem. Since this problem is highly ill-posed, many priors have been proposed to model the images and kernels [24–26]. However, most of these priors only perform well on generic natural images, but cannot generalize to specific image domains, like face [27], text [28] and low-illumination images [29]. Therefore, some priors (*e.g.*L_0-regularized intensity and gradient prior [30], face exemplars [31]) have been developed to handle these domain-specific image deblurring problems. Recently, some learning-based approaches have been proposed for blind image deblurring [27, 32, 33]. CNN-based models can handle more complex blur types and have enough capacity to train on large-scale datasets. Meanwhile, the Generative Adversarial Networks (GAN) have been found to be effective in generating more realistic images. Nonetheless, most of these methods need paired training data, which is expensive to collect in practice. Although numerous blur generation methods have been developed [32,34,35], they are not capable of learning all types of blur variants in the wild. Moreover, strong supervision may cause algorithms to overfit training data and thus cannot generalize well to real images.

In Chapter 5, we present an unsupervised method for domain-specific single-image deblurring based on disentangled representations. The disentanglement is achieved by splitting the content and blur features in a blurred image using content encoders and blur encoders. We enforce the KL divergence loss to regularize the distribution range of extracted blur attributes such that little content information is contained. Meanwhile, to handle the unpaired training data, a blurring branch

and the cycle-consistency loss are added to guarantee that the content structures of the deblurred results match the original images. We also add an adversarial loss on deblurred results to generate visually realistic images and a perceptual loss to further mitigate the artifacts. We perform extensive experiments on the tasks of face and text deblurring using both synthetic datasets and real images, and achieve improved results compared to recent state-of-the-art deblurring methods.

1.6 Regularized Metric Adaptation for Unconstrained Face Verification

In addition to the covariates like image blur and pose variations, for unconstrained face verification, there exists intrinsic domain mismatch between training and test data—the subjects in the training and test set are required to be mutually exclusive. This requirement often results in the model learned by training subjects performing poorly on test subjects. To build a connection between these two domains, we are inspired by the idea of one-shot learning [36]. The main idea of one-shot learning is to learn a discriminative model by using the test data and training data simultaneously. Wolf *et al.* [36] proposed the one-shot similarity (OSS) kernel based on a set of pre-selected reference images that are mutually exclusive to the pair of images being compared and trained a discriminative classifier between test images and the reference set. Guo *et al.* [37] followed the same rationale and developed the one-shot similarity approach based on partial least square regressors to leverage the rich information of the high-dimensional feature obtained by con-

catenating Gabor [38], LBP [39], and HOG [40] features. Crosswhite *et al.* [41] developed a one-shot similarity framework based on linear support vector machines and deep convolutional features of faces and achieved competitive results for the unconstrained face verification task.

In Chapter 6, we propose a metric adaptation method for unconstrained face verification. A template-specific metric is trained to adaptively learn the discriminative information in test templates and the negative training set, which contains subjects that are mutually exclusive to subjects in test templates. The proposed regularized joint Bayesian metric learning framework not only alleviates the overfitiing problem but also provides a way to efficiently reduce the model size. We also analyze the selection of the compact and representative negative set to speed up the training time and to reduce storage space. Experiments on the two challenging unconstrained face datasets yield promising results.

1.7 Contributions

- In Chapter 2, we comprehensively study the effects of seven covariates on the performance of unconstrained face verification.
  - We test seven covariates using state-of-the-art deep models. This gives insights into the limitations of many existing DCNNs for face covariates.
  - We study the mixed effects of multiple covariates. This is an important problem for unconstrained face verification yet not deeply explored by previous studies.

  - We propose to utilize gender information to curate the training data and achieve enhanced performance.

- In Chapter 3, we propose a pose-robust metric learning method to mitigate the performance drop induced by pose variation.
  - We present a novel metric learning approach for unconstrained face verification, and derive an optimization algorithm. The learned metric is robust to pose variations by reducing the effect of pose-sensitive information from the competing task.
  - We show that the proposed method yields promising experimental results on three challenging face datasets.

- In chapter 4, we propose an incremental dictionary learning approach for unsupervised domain adaptation.
  - We present a method to iteratively select and add supportive samples to the source domain to reduce the domain shift between source and target domains.
  - We design a stopping criterion to guarantee that the domain mismatch decreases monotonically during adaptation.
  - Experimental results on blurredface datasets and object classification tasks show that the proposed approach performs better than many state-of-the-art methods.

- In chapter 5, we propose a unsupervised method for domain-specific single-image deblurring.
  - We present an approach that uses disentangled representation and GANs for unsupervised image deblurring.
  - We significantly outperform other unsupervised deblurring methods and demonstrate superiority over supervised methods.
- In chapter 6, we propose a metric adaptation method for unconstrained face verification.
  - We present a one-shot learning-based method to improve the performance of unconstrained face verification.
  - We enforce a regularization term that reduces the model size.

Chapter 2: An Experimental Evaluation of Covariates Effects on Unconstrained Face Verification

2.1 Overview

Due to the recent development of DCNNs, face verification performance has significantly improved and has surpassed human performance in most controlled situations and some unconstrained cases [1–3]. Although deep features have proven to be more robust to moderate variations in pose, aging, occlusion and other factors than hand-crafted features, some recent works [7–10] have noticed that face verification performance is still significantly affected by covariates, which are factors that usually have an undesirable influence on face verification performance. The motivation for studying the effects of these covariates can be summarized as follows. First, we can better understand the fundamental challenges in face verification and the limitations of current algorithms. Second, the experimental results could provide insights for improving existing face verification algorithms.

In this chapter, we investigate two important covariate-related problems: a) how different covariates affect the performance of state-of-the-art DCNNs for unconstrained face verification; b) how to utilize covariate information to improve face

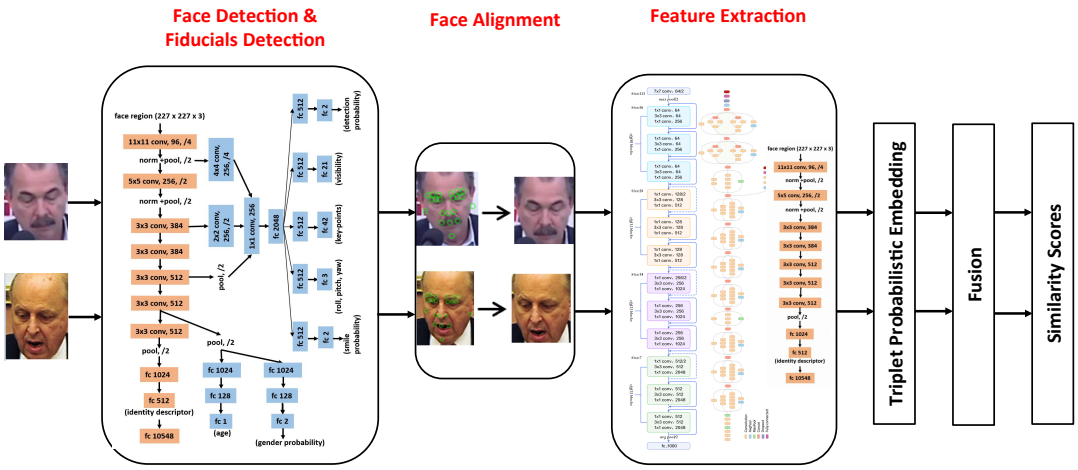


Figure 2.1: System pipeline for unconstrained face verification.

verification performance. For the first problem, we implement five state-of-the-art face DCNNs and evaluate them on three challenging covariate protocols: 1:1 covariate protocol of the IARPA JANUS Benchmark B (IJB-B) dataset [42] and its extended version, the IARPA JANUS Benchmark C (IJB-C) [43], and Celebrity Frontal-Profile (CFP) Face datasets [11]. IJB-B and IJB-C 1:1 covariate protocol are large-scale covariate dataset where seven covariates are evaluated. The CFP dataset mainly focus on pose variations. For the second problem, we utilize gender information to curate a noisy large-scale face dataset. Specifically, we leverage gender information to curate the training set and remove subjects mixed with images of both males and females. After retraining the model using the curated data, the verification performance improves at low FARs.

2.2 Evaluation Pipeline Overview

In this section, we briefly introduce the five deep networks used to perform unconstrained face verification over covariates. Before feeding a face image into these networks, preprocessing steps including face detection, facial landmark detection and face alignment are performed by using the multi-task CNN framework proposed in [44]. More details about the multi-task CNN are provided in Section 2.2.1.5. After feature extraction, we applied Triplet Probabilistic Embedding (TPE) [45] on the deep features to further improve the face verification performance. The TPE learns a projection matrix $\mathbf{W}$ by minimizing the negative log-likelihood objective function. More details can be found in [45]. The end-to-end system pipeline is illustrated in Figure 5.2.

2.2.1 Deep Representations for Faces

To capture the different characteristics of faces, we use features extracted from five state-of-the-art deep neural networks. These five networks have different architectures and training sets with their own strengths and weaknesses.

2.2.1.1 Training set preparation

To train the deep networks, we use UMD-Faces [46, 47], Megaface [48], and MS-Celeb-1M [15]. In addition, we found that directly using the original MS-Celeb-1M dataset for training does not achieve good performance because the labels are very noisy. Therefore, we used a curated version of MS-Celeb-1M dataset which is

done using a clustering method introduced in [16]. The curated dataset contains about 3.7 millions face images from 57,440 identities. After curation, many noisy labels are removed while sufficient amount of face images with different variations are retained.

2.2.1.2 CNN-1

This network employs the ResNet-27 model introduced in [49]. We modify the original model by removing the center loss and replacing the softmax loss with the L_2-softmax loss introduced in [10]. In addition, we also add one more fully connected layer with 512-D before the L_2-softmax layer to reduce the feature dimension and the total number of model parameters. For the input size, we change the original size of 112×96 to 128×128 for improved face alignment. To train the model, we use a curated version of the MS-Celeb-1M dataset described in Section 2.2.1.1, which contains 3.7 million images from $57,440$ subjects.

2.2.1.3 CNN-2

The second network uses the ResNet-101 [50] architecture as the base network. CNN-2 is deeper than CNN-1 and accepts larger inputs of dimensions 224×224. The basic blocks for CNN-2 use bottleneck structures to reduce the number of model parameters and achieve deeper networks given certain memory constraints. Similar to CNN-1, CNN-2 also replaces the original softmax loss with the L_2-softmax loss and adds an additional fully connected layer before the L_2-softmax layer. CNN-2 is

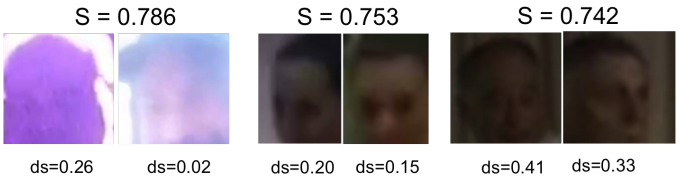


Figure 2.2: Examples of hard negative pairs with low detection confidence but high similarity scores. *ds* indicate the detection scores for the images and S represents similarity score for each pair.

trained using two different training sets and thus two different models are obtained. One model is called CNN-2_S because a small training set is used (curated MS-Celeb-1M dataset) and the other model is called CNN-2_L because it uses a larger training set (curated MS-Celeb-1M dataset, about 300,000 still images from the UMDFaces dataset [46], and about 1.8 million video frames from the UMD-Faces Video dataset [47]).

2.2.1.4 CNN-3

The Inception-ResNet-v2 [51] model is used as the base network. This model combines the inception architecture with residual connections and scaling layers which scale down the residuals for more stable training. We also add a 512-D fully connected layer before the last layer. The training set is the same as for CNN-2.

2.2.1.5 CNN-4

This network is based on the all-in-one CNN architecture [44]. The model is trained in a multi-task learning framework which utilizes the correlations among different tasks to learn a more robust model than learning each task individually. The face detection and facial landmark detection branches share the first six layers and have two separate fully connected layers for each task. The face recognition branch consists of seven convolutional layers followed by three fully connected layers. In this chapter, we mainly utilize the face detection, facial landmark detection branches for face alignment, and the face recognition branch to generate face features. We also use the gender classification branch to estimate gender probabilities. The same training set used for CNN-1 and CNN-2_S is used for this network.

2.2.2 Face Matching and Score Level Fusion

After we obtain the extracted features from the learned deep networks and the embedding matrix $\mathbf{W}$ from TPE [45], the similarity scores for each pair $\{x_i, x_j\}$ is computed by simply using the cosine similarity of the two embedded features:

$$s_{ij} = \frac{(\mathbf{W}x_i)^T(\mathbf{W}x_j)}{\|\mathbf{W}x_i\| \, \|\mathbf{W}x_j\|} \tag{2.1}$$

In the last stage of the proposed system, we fuse the scores computed from the five networks as the final similarity score. We observe that the similarity scores may become unreliable when the image quality is poor. Meanwhile, we find the face detection score obtained from the face detection branches of the CNN-4 is a

good indication of image quality. Figure 2.2 shows some hard negative pairs with low detection scores but high similarity scores. We notice that the main reason for the high similarity scores is that these pairs are all very blurred and each pair has similar background. To address this issue, we reweight the similarity scores when the face detection scores of the corresponding pairs are low.

$$\hat{s_i} = \begin{cases} s_i, & \text{if } ds > thr \\ \alpha s_i, & \text{otherwise,} \end{cases} \tag{2.2}$$

where ds is the minimum of the detection scores for the pair of faces, thr is the threshold, α is the reweight coefficient.

Then we simply average the reweighted similarity scores from the five networks to get the final results.

$$s = \frac{1}{5} \sum_{i}^{5} \hat{s}_i \tag{2.3}$$

2.3 Performance Improvement by Exploiting Gender Information

Although many noisy labels are removed after curating the training set using the clustering method mentioned in Section 2.2.1.1, there still exists many noisy labels which cannot be handled by clustering. Moreover, we observe that some clusters are even mixed with different genders. This motivates us to further curate the training set by exploiting the gender information. First, gender probabilities are estimated using the all-in-one CNN network [44] for all the face images in the pre-curated MS-Celeb-1M dataset in 2.2.1.1. Since gender estimation may become

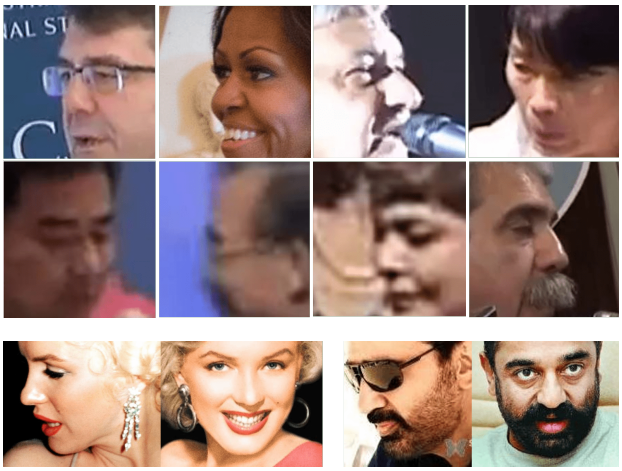


Figure 2.3: Sample images for IJB-B (first row), IJB-C (second row) and CFP (third row) datasets.

unreliable when gender probabilities are near 0.5, we only consider faces with gender probability greater than 0.6 (male) or smaller than 0.4 (female). For each subject, if the number of faces from the minority gender is more than 3% of the total number of faces, we eliminate the whole subject. In total, we removed 248,059 faces from 4,160 subjects. It is worth mentioning that we also tried other possible criteria for gender-based curation (*e.g.*, only removing images from minority gender, or use other thresholds instead of 3%) but observed a drop in performance.

2.4 Experimental Results

To analyze the covariate effects on unconstrained face verification performance, we evaluate the five deep networks on three challenging face datasets that have face verification covariate protocols: the IARPA JANUS Benchmark B (IJB-B) 1:1 covariates [42], the IARPA JANUS Benchmark C (IJB-C) 1:1 covariates [43] and the

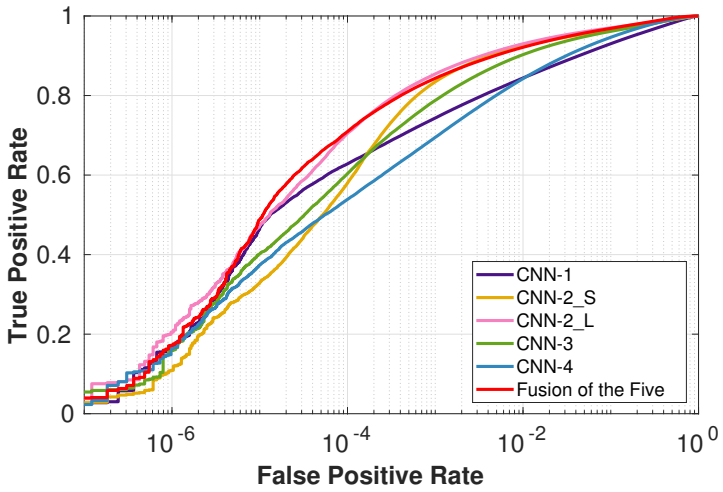
Celebrities in Frontal-Profile in the Wild (CFP) [11]. The IJB-B and IJB-C 1:1 covariates both contain seven covariate protocols while the CFP dataset mainly focuses on extreme pose variations. For IJB-B and IJB-C, we first report the performance of each individual network on the overall protocol, and then use the score-level fusion method to analyze each covariate.

Method	TAR@FAR = 10^{-7}	TAR@FAR = 10^{-6}	TAR@FAR = 10^{-5}	TAR@FAR = 10^{-4}	TAR@FAR = 10^{-3}	TAR@FAR = 10^{-2}	TAR@FAR = 10^{-1}
IJB-B before curation	**0.0252**	0.1602	0.4455	0.6282	0.7474	**0.8493**	**0.9328**
IJB-B after curation	0.0245	**0.1731**	**0.4636**	**0.6284**	**0.7481**	0.8447	0.9290
IJB-C before curation	0.2417	0.3596	0.5023	0.6403	0.7660	**0.8624**	**0.9368**
IJB-C after curation	**0.2661**	**0.3946**	**0.5378**	**0.6586**	**0.7684**	0.8586	0.9337

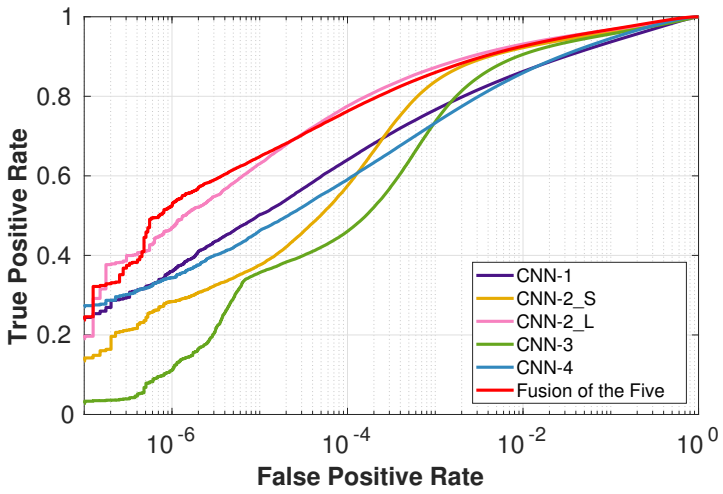
Table 2.1: Performance comparison between before and after gender-based training set curation on IJB-B and IJB-C 1:1 covariate protocol. All the results are generated using the CNN-1 architecture.

Method	TAR@FAR = 10^{-7}	TAR@FAR = 10^{-6}	TAR@FAR = 10^{-5}	TAR@FAR = 10^{-4}	TAR@FAR = 10^{-3}	TAR@FAR = 10^{-2}	TAR@FAR = 10^{-1}
VGG-Face	0.0150	0.0440	0.0994	0.1515	0.2190	0.3318	0.5723
Center-Face	0.0063	0.0353	0.0780	0.1363	0.2370	0.4206	0.7501
Center-Face(retrain)	**0.0517**	0.1656	0.3880	0.6014	0.7620	0.8692	0.9460
Fusion of our five model	0.0396	**0.1707**	**0.4882**	**0.7093**	**0.8434**	**0.9213**	**0.9688**

Table 2.2: Performance comparison for different methods on the IJB-B 1:1 covariate overall protocols. Our fusion results are generated by detection score-based fusion of the five deep models. VGG-Face and Center-Face results are derived by applying their pretrained models to extract features and following the IJB-B 1:1 covariate overall protocol. Center-Face(retrain) is retrained using the curated MS-Celeb-1M dataset and the Center-Face model.



(a) ROC curves for IJB-B 1:1 covariates



(b) ROC curves for IJB-C 1:1 covariates

Figure 2.4: ROC curves for IJB-B and IJB-C 1:1 covariates overall protocols without specifying separate covariate labels. The fusion results are obtained by detection-score based fusion of the five CNN networks. The figures are best viewed in color.

Method	TAR@FAR = 10^{-7}	TAR@FAR = 10^{-6}	TAR@FAR = 10^{-5}	TAR@FAR = 10^{-4}	TAR@FAR = 10^{-3}	TAR@FAR = 10^{-2}	TAR@FAR = 10^{-1}
VGG-Face	0.0513	0.0792	0.1159	0.1616	0.2275	0.3396	0.5918
Center-Face	0.0479	0.0652	0.1005	0.1629	0.2746	0.4739	0.7733
Center-Face(retrain)	**0.2417**	0.3596	0.5023	0.6403	0.7660	0.8624	0.9368
Fusion of our five model	0.2371	**0.5249**	**0.6478**	**0.7623**	**0.8599**	**0.9261**	**0.9681**

Table 2.3: Performance comparison for different methods on the IJB-C 1:1 covariate overall protocol. Our fusion results are generated by detection score-based fusion of the five deep models. VGG-Face and Center-Face results are derived by applying their pretrained model to extract features and following the IJB-C 1:1 covariate overall protocol. Center-Face(retrain) is retrained using the curated MS-Celeb-1M dataset and the Center-Face model.

2.4.1 IJB-B and IJB-C 1:1 covariate protocol

The IARPA JANUS Benchmark B (IJB-B) dataset [42] is a moderate-scale unconstrained face dataset with face detection, recognition and clustering protocols. It consists of 1845 subjects with human-labeled ground truth face bounding boxes, eye/nose locations, and covariate meta-data such as occlusion, facial hair, and skin tone for 21,798 still images and 55,026 frames from 7,011 videos. The 1:1 covariate protocol of IJB-B aims to analyze the effects of seven different covariates (i.e., pose (yaw and roll), age, facial hair, gender, indoor/outdoor, occlusion (nose and mouth visibility, forehead visibility), and skin tone.) on face verification performance. The protocol has 20,270,277 pairs of templates (3,867,417 positive and 16,402,860 negative pairs) which enables us to evaluate algorithms at low FAR region of ROC curves (*e.g.*, FAR at 10^{-5} and 10^{-6}). Each template contains only one

image or a video frame. The IARPA JANUS Benchmark C (IJB-C) dataset [43] is an extended version of the IJB-B dataset, which consists of 3,531 subjects containing 140,739 images and video frames. The 1:1 covariate protocol has 47,404,001 pair of templates (7,819,362 positive and 39,584,639 negative pairs). Some sample images of the IJB-B and IJB-C datasets are shown in Figure 2.3.

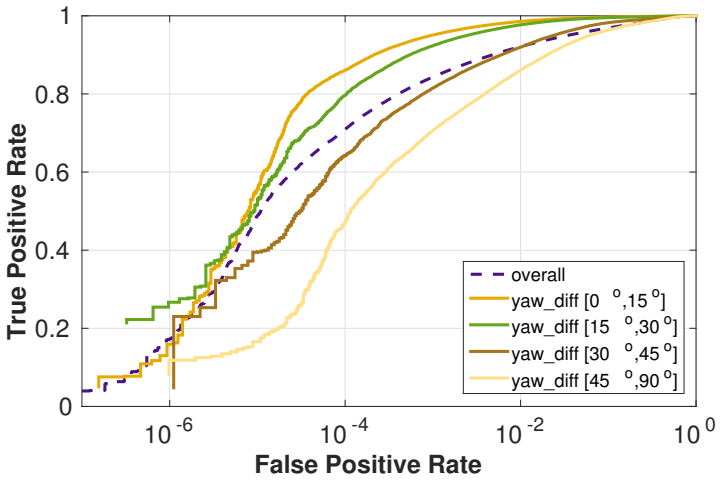
To understand the effects of different covariates on face verification performance, in addition to the identity label (positive or negative) for each pair of templates, covariate labels are also assigned to each pair. To analyze a certain covariate (like gender), all pairs are split into groups based on the value of covariate labels (female, male). The ROC curves are drawn for each group and the performance difference among different groups reflects the effects of the covariates. When we evaluate the general performance of an algorithm, all the pairs are mixed together without their specifying separate covariate labels.

2.4.2 Evaluation on the overall protocol

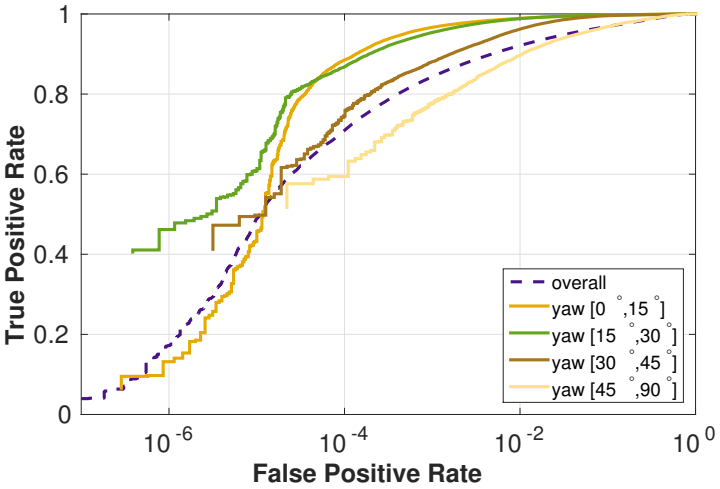
In the following sections, we first present our experimental results on the overall protocol where covariate labels are not involved and then delve into the details of each covariate result.

2.4.2.1 Results for five deep networks and score-level fusion

To compare the performance of five deep networks, we present the ROC curves for each network and their score-level fusion. For detection score-based fusion, threshold thr is set to 0.75 and the reweighting coefficient α is set to 0.8. We also



(a) ROC curves with yaw difference changes for IJB-B



(b) ROC curves with absolute yaw angle changes for IJB-B

Figure 2.5: ROC curves (a) when the yaw difference between two face images changes and (b) when absolute yaw angle of faces changes. The range is from 0° to 90° because we average the features for original face and its mirrored image as the final face representation. The absolute yaw angles are computed by averaging two faces. The dashed line represents the results for the overall protocol.

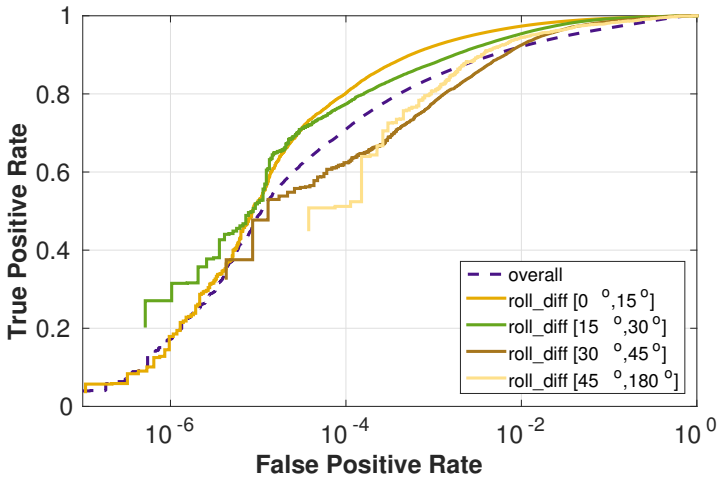


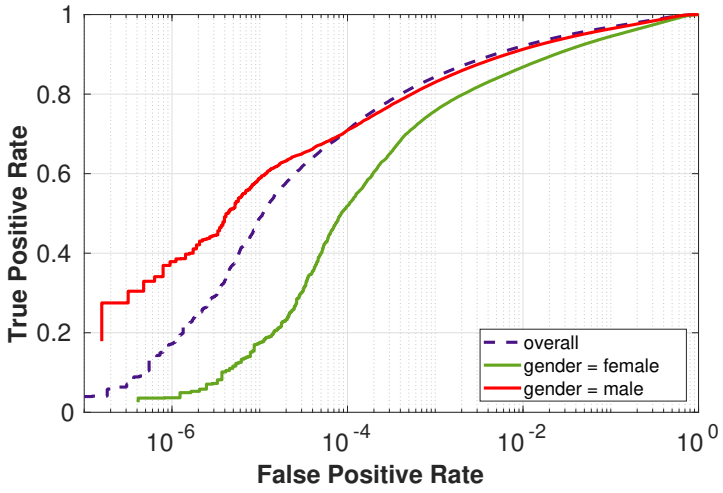
Figure 2.6: ROC curves when the roll angle difference between two face images changes for IJB-B. The range is from 0° to 180°. The dashed line represents the results for the overall protocol.

did a sensitivity analysis on these two parameters, the details of which are included in [23]. Figures 2.4(a) and 2.4(b) show the performance for IJB-B and IJB-C 1:1 covariates respectively. From these figures, we observe that CNN-2_S and CNN-3 perform very well at high FARs of the ROC curve, but the performance drops rapidly at low FARs. In contrast, CNN-1, and CNN-4 have smoother curves and perform better at low FARs but worse at high FARs. Meanwhile, CNN-2_L shows very strong performance for all FARs and outperforms the other four networks in middle range of FARs (FAR=10^{-4}, 10^{-3}). Moreover, the fusion results of the five networks outperform all individual models, especially at low FAR of the ROC curve for the IJB-C dataset. This demonstrates the complementary behavior of the different models and fusion can always yield some improvements over individual models. By

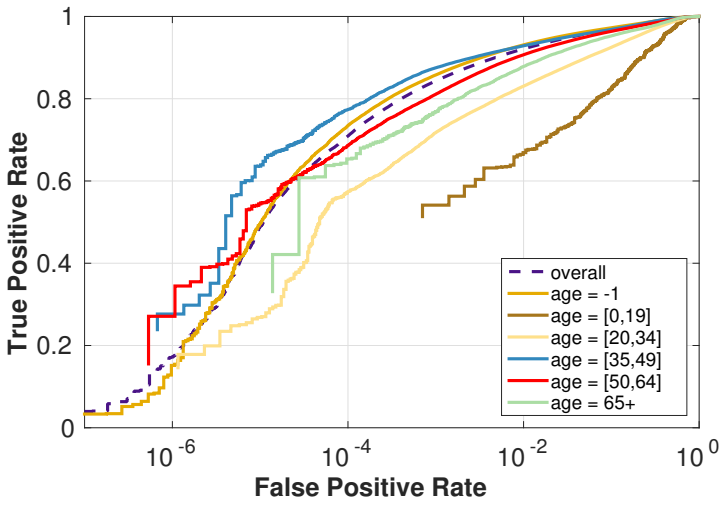
comparing the ROC curves of IJB-B and IJB-C datasets, we can see similar trends when FARs are larger than 10^{-4} but the performance for IJB-B drops faster at low FARs of the ROC curve. In addition, at low FARs, different algorithms perform very differently for IJB-C but similarly for the IJB-B dataset. This indicates that the IJB-B dataset contains more hard negative pairs.

2.4.2.2 Performance improvement by gender based training set curation

To test the effectiveness of the dataset curation method discussed in Section 2.3, we retrain CNN-1 using the training set curated by exploiting gender information and compare with results obtained before curation. From Table 2.1 it can be seen that the performance is improved at low FARs of ROC curves after training set curation on both IJB-B and IJB-C datasets. Since the goal of gender-based curation is to improve the model's capability to distinguish male and female subjects who looks very similar, performance improvements at low FARs are consistent with this goal because it indicates that the model can deal with hard negative pairs in a better way. On the other hand, we notice that the performance improvements on IJB-C are larger than on IJB-B, which means the gender information is more useful to detect the hard negative pairs in IJB-C than in IJB-B.



(a) ROC curves with different genders for IJB-B



(b) ROC curves with age changes for IJB-B

Figure 2.7: ROC curves for different genders and for the case of age variation. The dashed line represents the results for the overall protocol. Ages that are different for two images in a pair are labeled as -1.

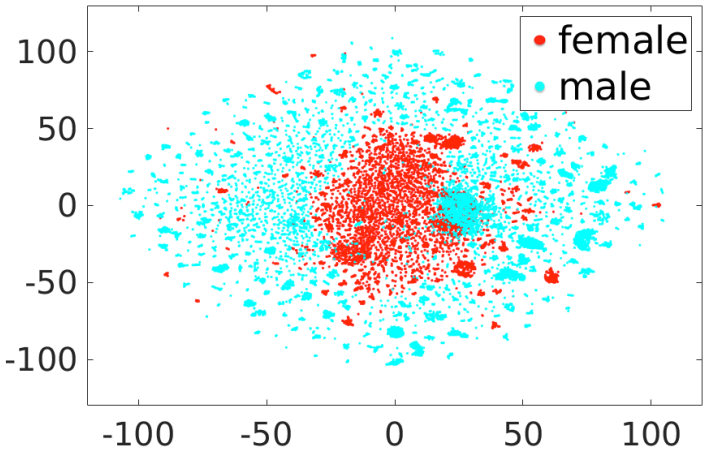


Figure 2.8: t-SNE visualization of CNN_2L features from different genders for IJB-B dataset. Blue dots indicate males and red dots represent females.

2.4.2.3 Comparisons with other competitive methods

We also compare our fusion results with some other state-of-the-art methods and two widely used public models are considered: VGG-Face [52] and Center-Face [49]. We used the pretrained models provided by authors to extract features and followed their preprocessing steps on face images. As shown in Table 2.2 and Table 2.3, our fusion results outperform both VGG Face and Center-Face by large margins. There are two main reasons for this dramatic performance difference. First, we employ deeper models and various architectures to capture different characteristics of faces and conduct score-level fusion to further boost the performance. Second, the training set we use contains more faces with diverse face variations. In order to investigate the effect of using different training sets, we retrain the Center-Face model using the curated MS-Celeb-1M dataset. As illustrated in Table 2.2

and Table 2.3, we see significant improvements in performance compared to the pretrained model, but the proposed fusion method still outperforms the retrained model significantly.

2.4.3 Evaluation on pose

To evaluate the effects of pose variations on face verification performance, the protocol provides yaw and roll angles for each face. Since we use the average of the features for original face and its mirrored version as the final face representation, this restricts the range of yaw to $[0°, 90°]$ and roll to $[0°, 180°]$. Based on the yaw difference between a pair of faces, we divide all pairs into four groups: $[0°, 15°]$, $[15°, 30°]$, $[30°, 45°]$, and $[45°, 90°]$. Similarly, pairs are also divided into four groups based on roll difference: $[0°, 15°]$, $[15°, 30°]$, $[30°, 45°]$, and $[45°, 180°]$. We did not include the IJB-C plots here because they show similar results as IJB-B.

From Figure 2.5(a), we observe that the yaw difference between a pair of faces significantly affects the face verification performance. The ROC curves decrease monotonically as the yaw difference between the two faces increases. Moreover, the performance drops much faster when the yaw difference is larger than 30°. This supports the following two findings: a) deep face representations are robust to moderate yaw changes (less than 30°); b) the state-of-the-art deep networks are still sensitive to large yaw variations (larger than 30°). However, when considering the low FARs regoins, we find the performances for different groups become similar. In addition to yaw difference between two faces, another key factor that may influence

the performance is the absolute yaw value of faces. In other words, even if the yaw difference between two faces is relatively small (less than 15°), the performance may still be affected when the absolute yaw angles for both faces are large. In order to separate this factor from that due to yaw difference, we further split the group of yaw difference $[0°, 15°]$ into four subgroups based on their absolute yaw angles: $[0°, 15°]$, $[15°, 30°]$, $[30°, 45°]$, and $[45°, 90°]$, where the degrees are computed by averaging the absolute yaw angles of a pair of faces. The ROC curves are shown in Figure 2.5(b). Similar to the effect of yaw difference, the absolute yaw angles of faces larger than 30° cause a large performance drop while performance is not affected much when yaw angles are less than 30°. By comparing Figures 2.5(a) and 2.5(b), we have another interesting finding: performance for absolute yaw angles in $[45°, 90°]$ and for yaw difference in $[45°, 90°]$ are comparable, which means that as long as at least one of the two faces is in extreme yaw angle, the performance will be poor. This result demonstrates that face images with extreme yaw angles ($[45°, 90°]$) are hard for face matching regardless of the yaw difference because a large part of facial information is missing.

Figure 2.6 shows the face verification performance for various roll difference between two faces. We find that performance is better for groups whose roll differences are smaller than 30°. This result is surprising because in general the roll difference should not affect the face verification performance since 2D face alignment is performed before face matching to normalize all faces to have the same roll angle. However, the performance drop when increasing the roll difference shows that facial landmarks may not be accurate so that faces are not normalized as expected when

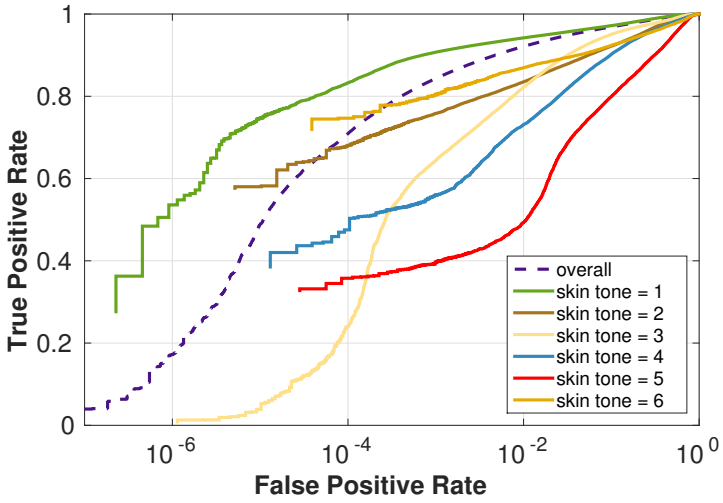
the roll angle is large.

2.4.4 Evaluation on gender

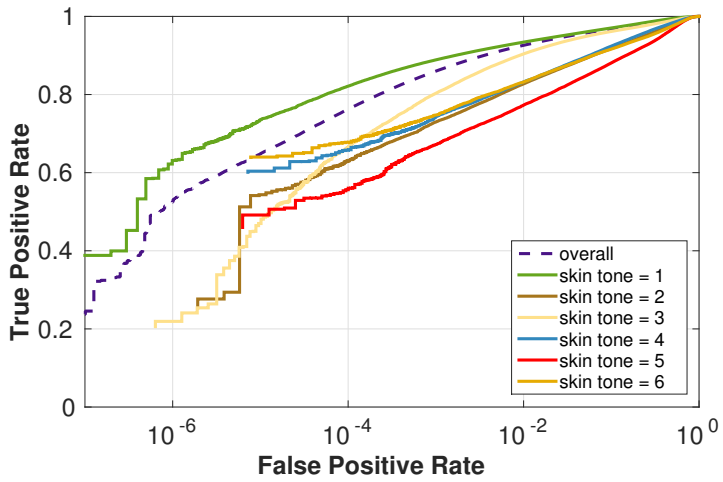
From Figure 2.7(a), it can be observed that the performance for men is much better than women on the IJB-B dataset. The results for the IJB-C dataset show similar trends and are not included. A possible explanation for this result is that women's faces are often occluded by their long hair and their face appearance are changed by makeup. To further investigate the underlying reasons of our observation, we use t-SNE plots [53] to analyze the feature distributions under different genders and the results are illustrated in Figure 2.8. The small clusters represent different subjects and we also include the t-SNE visualization based on identities in [23]. We can see that the feature distributions for men are much more separated and discriminative than women, which lead to better performance.

2.4.5 Evaluation on age

The 1:1 covariate protocol labels the test pairs into six categories based on their age distributions. Ages that are different for two faces in a pair are labeled as -1. Results for IJB-B dataset are shown in Figure 2.7(b). We do not include the IJB-C plots here because they show similar results as for IJB-B. The dashed line represents performance for the overall protocol while the solid lines present curves for different age groups. It is shown that performance goes up when age increases from 0 to 49 and begins to drop when the age is higher than 49. It means the

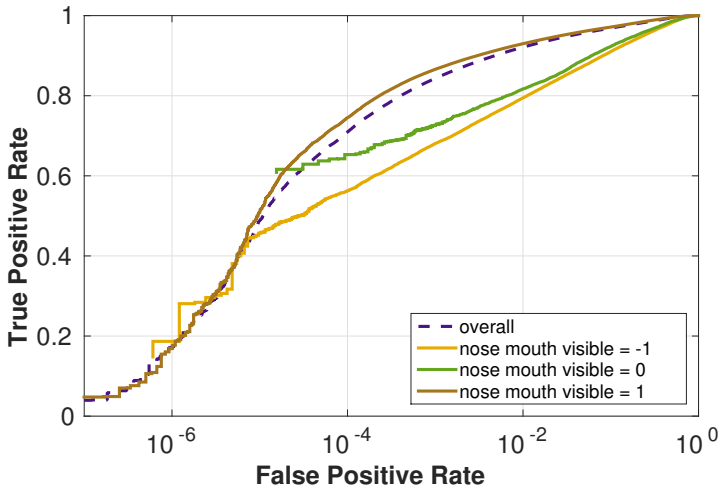


(a) ROC curves with skin tone changes for IJB-B

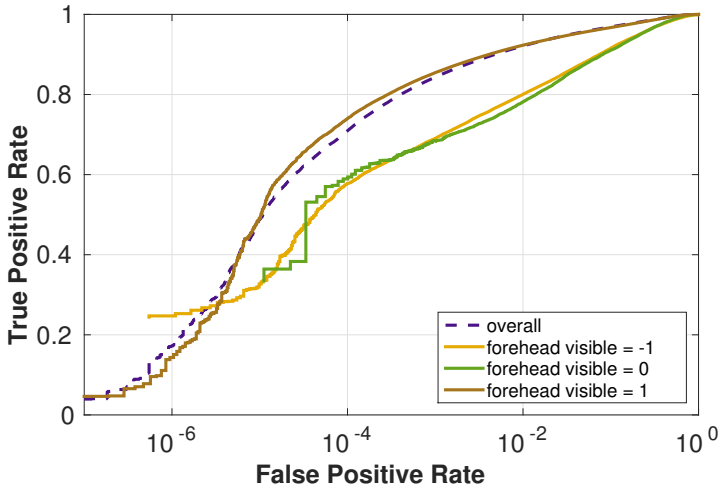


(b) ROC curves with skin tone changes for IJB-C

Figure 2.9: ROC curves with changes in skin tone. The dashed line represents performance for the overall protocol while the solid lines are curves for different skin tones. light pink, light yellow, medium pink/brown, medium yellow/brown, medium-dark brown and dark brown are labeled as 1, 2, 3, 4, 5, 6 respectively.

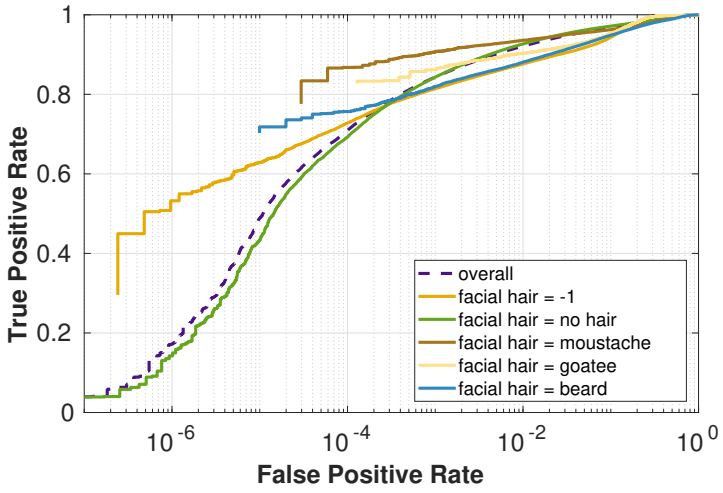


(a) ROC curves with nose/mouth visibility changes

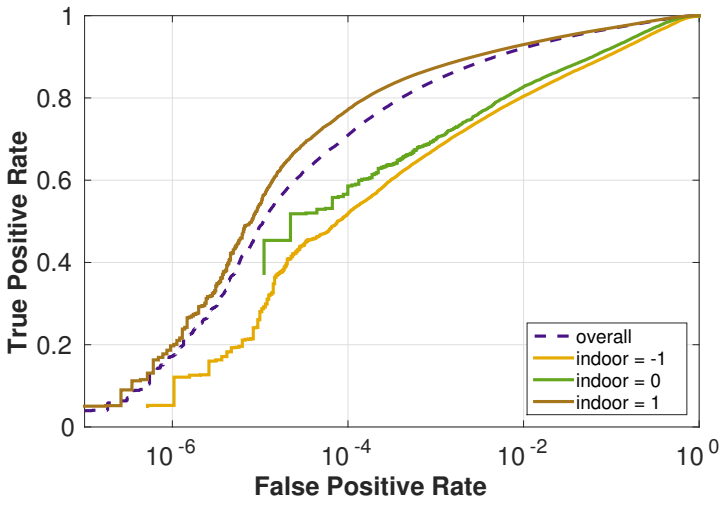


(b) ROC curves with forehead visibility changes

Figure 2.10: ROC curves corresponding to nose/mouth and forehead visibilities for IJB-B dataset. label 0 represents non-visible and label 1 means visible.



(a) ROC curves with facial hair changes for IJB-B



(b) ROC curves with indoor/outdoor changes for IJB-B

Figure 2.11: ROC curves for varying facial hairs and for indoor/outdoor. For indoor/outdoor, outdoor is labeled as 0 and indoor is 1. Label -1 means one image is taken indoor and the other outdoor.

middle-age group ([35, 49]]) is the easiest one to be recognized while too young or too old subjects are both challenging for face verification. One possible explanation for this result may be because new born babies all look very similar and their unique facial features begin to emerge as they grow. However, as people age, some common features for elderly people like wrinkles and sagging skins impair the uniqueness of their facial characteristics, which may make them harder to be distinguished. On the other hand, we find the performances for age groups that are older than 35 become closer at low FARs. In addition, we notice that age group -1 (ages of two images are different.) performs similarly as the overall protocol, which means cross-age face verification is as hard as the general case. Nonetheless, this dataset does not fully explore the difficulty of cross-age face verification because the IJB-B and IJB-C datasets do not have images from the same person across large age gaps.

2.4.6 Evaluation on skin tone

For skin tone, the protocol defines six classes: (1) light pink, (2) light yellow, (3) medium pink/brown, (4) medium yellow/brown, (5) medium dark brown, and (6) dark brown. From Figure 2.9, we observe that the performances for different skin tone groups show different trends on IJB-B and IJB-C. For IJB-B, the ROC curves for different groups are well separated. A general trend is that the performance drops when the skin tone becomes darker. However, a counterexample is skin tone group 6 (darkest), which performs better than group 2 to group 5. On the other hand, the performance for group 3 drops rapidly and performs the worst at low

FARs. This demonstrates that the hard negative pairs for group 3 are more difficult to recognize. For IJB-C, except group 1 and group 5 which have the same trends as IJB-B, the performances for other skin tone groups are very close. Thus, we can only draw the conclusion that skin tone group 1 is the easiest and skin tone group 5 is the hardest for face verification. However, since defining or recognizing skin tones is ambiguous sometimes, it is hard to decide which skin tone is easier for face verification only from these results. In Figure 2.12, we visualize the feature distribution for different skin-tone groups in the IJB-B dataset. We can easily find that features for group 1 (shown in red dots) are most separated and thus achieve the best performance. Nonetheless, feature distributions for other groups do not show much information.

2.4.7 Evaluation on mouth and nose, and forehead visibility

To evaluate the effects of occlusion, the protocol tests two types of visibilities: mouth and nose visibility, and forehead visibility. Label 0 (1) represents the parts are both invisible (visible) for two images, and label -1 means the part is visible for one image but not for the other. The ROC curves for the IJB-B dataset are presented in Figures 2.10(a) and 2.10(b) respectively. We see similar results for mouth/nose and forehead visibility: class -1 and 0 have comparable performance but are worse than class 1, which means that performance falls by large margins if nose, mouth or forehead are occluded for at least one of the images. This result indicates the importance of the visibility of key facial parts for recognizing faces.

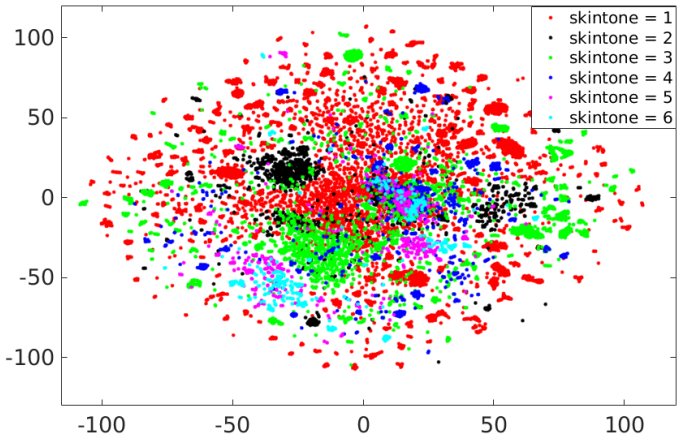


Figure 2.12: t-SNE visualization of CNN_2L features from different skin tones for IJB-B dataset.

However, when considering the low FARs regoins, we find the performances for different groups become similar. This means for low FAR regions, occlusion is not the key factor that decides performance since the pairs are often affected by many covariates (*e.g.*, pose, occlusion, illumination).

2.4.8 Evaluation on facial hair

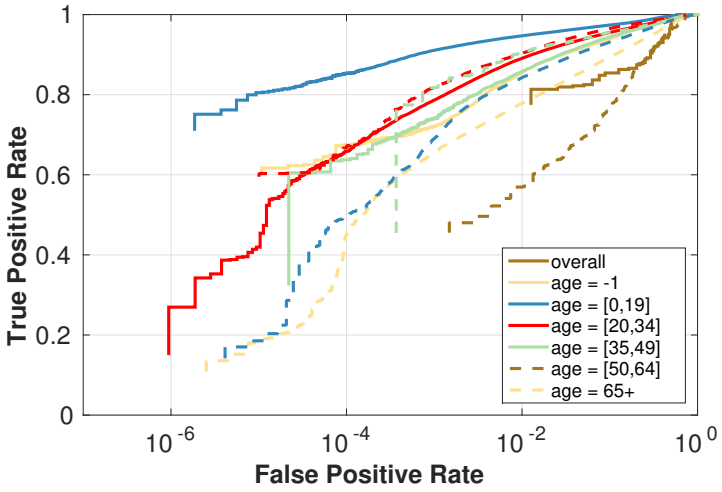
There are four classes for evaluation in facial hair protocol: no facial hair, moustache, goatee and beard respectively. Label -1 means facial hair classes are different for two images. From Figure 2.11(a), we observe that performance is not very sensitive to facial hair changes. This result demonstrates that facial hair does not change the key features of faces and state-of-the-art deep models can handle most facial hair variations.

2.4.9 Evaluation on indoor/outdoor

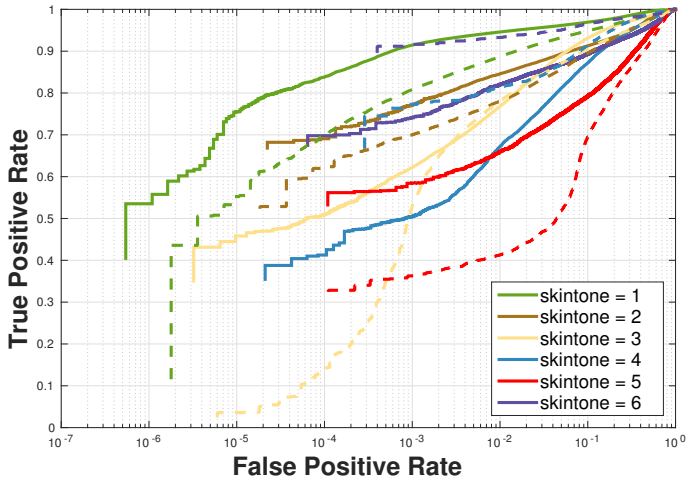
The last covariate we evaluate in the protocol is indoor/outdoor. Outdoor is labeled as 0 and indoor is 1. Label -1 means one image is taken indoor and the other outdoor. Performance is shown in Figure 2.11(b). We can see that the performance of class 1 is much better than class 0 and -1. This implies that indoor images are easier for face verification. Different from occlusion, we find that performance for indoor is still better than outdoor even at low FARs. This leads to a claim that indoor is an important condition to recognize hard negative pairs. There are two possible reasons for this result. First, outdoor images could be easily over-exposed and lose significant facial information. Second, outdoor images are often taken by hand-held cameras when people are walking. In contrast, indoor images are usually captured without much motion. So the image quality for indoor images is often better than outdoor images.

2.4.10 Evaluation on the effects of multiple covariates

In unconstrained face verification, multiple face covariates are often correlated with each other which may affect the performance. It has been found that some covariates may show different trends on face verification performance when other covariates are considered together [54, 55]. To study the correlations among the different covariates, we choose four pairs of related covariates and evaluated their interactive effects: gender and age, gender and skin tone, indoor (outdoor) and nose-mouth visibility, indoor (outdoor) and yaw angle difference. All experimental

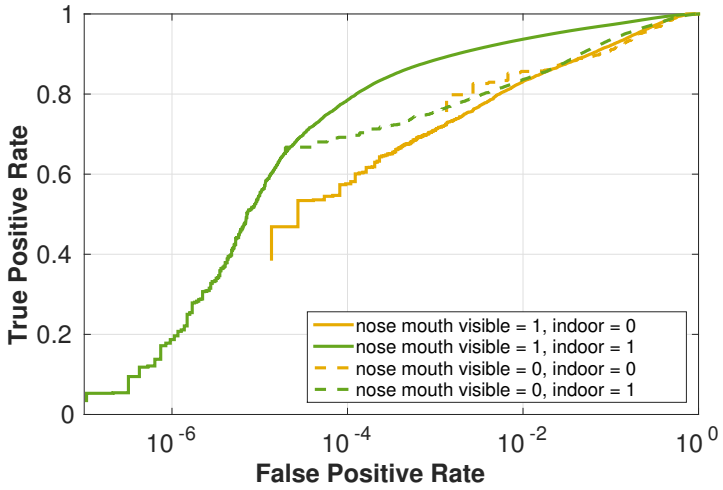


(a) ROC curves with age and gender changes on IJB-B

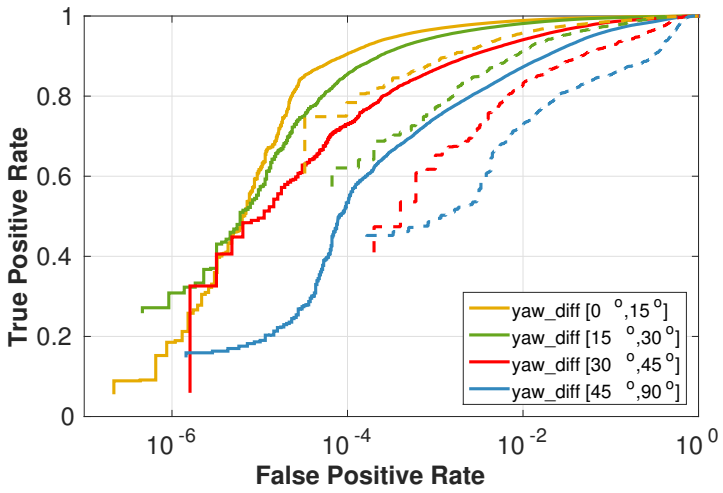


(b) ROC curves with skin tone and gender changes on IJB-B

Figure 2.13: ROC curves corresponding to age and gender (left) changes, and skin tone and gender (right) changes. Color lines represent different age groups and skin tones where small numbers represent light skin tones. Results for women are shown in dashed lines and solid lines represent results for men.



(a) ROC curves with indoor/outdoor and nose-mouth visibility changes.



(b) ROC curves with indoor/outdoor and yaw difference changes.

Figure 2.14: ROC curves corresponding to nose-mouth visibility and indoor/outdoor (left), and yaw difference and indoor/outdoor. Outdoor is shown in dashed lines and solid lines represent indoor.

results are reported only for the IJB-B dataset.

2.4.10.1 Evaluation on gender and age

In order to show how gender and age influence each other, we draw the ROC curves in Figure 2.13(a) for each possible combination of values from genders and age groups. Different age groups are represented using different colors and men/women is showed in solid/dashed lines. First, we fix the gender factor and compare the performance of different age groups for males or females. We see that males and females show very different trends on age group effects. More specifically, men in middle age group $[35, 49]$ performs best and the performances for men in age group $[50, 64]$ and 65+ decrease. In contrast, for women the performance always increases when age groups get older.

Alternatively, we can fix the age group factor and compare the performance of men and women for each age group. As observed in Section 2.4.4, in general, results for men are better than those for women. However, this finding does not hold for age groups $[50, 64]$ and 65+. For age group $[50, 64]$, men and women perform comparably while for age group 65+ women outperform men.

2.4.10.2 Evaluation on gender and skin tone

We repeated the procedure discussed above for analyzing the combination of gender and skin tone. The ROC curves are shown in Figure 2.13(b). For skin tone groups 4 and 6, the performance for women is better than that for men, while men perform better for group 1, 2 and 5. For skin tone group 3, men and women perform

similarly. This result shows that the combinations of gender and skin tone do not show clear trends and the performance is dependent on datasets.

	Frontal-to-Frontal			Frontal-to-Profile		
	Accuracy	EER	AUC	Accuracy	EER	AUC
Deep features [11]	0.964(0.007)	0.035(0.007)	0.994(0.003)	0.849(0.018)	0.150(0.020)	0.930(0.016)
Human [11]	0.962(0.007)	0.053(0.018)	0.982(0.011)	0.946(0.011)	0.050(0.011)	0.989(0.005)
CNN-1	0.988(0.002)	0.012(0.004)	0.999(0.001)	0.938(0.012)	0.062(0.013)	0.986(0.005)
CNN-2_S	**0.997(0.003)**	**0.003(0.003)**	**1.000(0.000)**	**0.981(0.007)**	**0.018(0.007)**	**0.997(0.002)**
CNN-2_L	0.996(0.003)	0.004(0.003)	**1.000(0.000)**	0.980(0.004)	0.021(0.006)	**0.997(0.002)**
CNN-3	0.994(0.004)	0.006(0.005)	**1.000(0.001)**	0.969(0.009)	0.029(0.011)	0.994(0.003)
CNN-4	0.982(0.008)	0.018(0.008)	0.998(0.001)	0.912(0.012)	0.085(0.012)	0.972(0.006)
Fusion	0.995(0.003)	0.004(0.004)	**1.000(0.001)**	0.973(0.006)	0.027(0.008)	0.996(0.002)

Table 2.4: Performance comparison for different methods on CFP dataset. Our fusion results are generated by averaging the four deep models.

2.4.10.3 Evaluation on indoor (outdoor) and nose-mouth visibility

In addition to the demographic covariates, we are also interested in the mixed effects of covariates related to extrinsic factors. Figure 2.14(a) shows the performance for different indoor/outdoor and nose-mouth visibility combinations. As we already saw, visible nose-mouth and indoor are more favorable for better performance. However, these two factors may not have independent impacts on performance. From Figure 2.14(a), we find that only when nose or mouth is visible and the images are taken indoor, the performance is good. Either occlusion or outdoor

can deteriorate the performance. At low FARs, we find that indoor/outdoor is more important than nose-mouth visibility, as the performance for green dashed line is better than yellow solid line in this region. This finding confirms the claim made in Section 2.4.7 and 2.4.9.

2.4.10.4 Evaluation on indoor (outdoor) and yaw angle difference

The last combination we considered is indoor/outdoor and yaw angle difference. The ROC curves are presented in Figure 2.14(b). We notice that when fixing the indoor/outdoor factor, the performance for smaller yaw angle difference is always better. On the other hand, when the yaw angle difference is fixed, indoor faces always outperform outdoor faces. This result demonstrates that yaw angle difference and indoor/outdoor can affect the face verification performance independently and changing any one of the two factors can affect the performance.

2.4.11 Evaluation on the CFP dataset

Since pose variation is a key challenging issue for face verification, we also used the Celebrities in Frontal-Profile (CFP) dataset to further investigate the underlying effects of extreme pose variations on unconstrained face verification performance. The CFP dataset consists of 7,000 still images from 500 subjects with 14 images per subject. For each subject, it has 10 images in frontal pose and 4 images in profile pose. To evaluate the performance for different poses, the protocol contains two settings: frontal-to-frontal (FF) and frontal-to-profile (FP) face verification. In the

frontal-to-frontal setting, two test images are both in frontal pose and in frontal-to-profile setting, a test pair includes one frontal face and one profile face. Each setting divides the whole dataset into ten splits and each split consists of 350 positive and 350 negative pairs. Some sample images are shown in Figure 2.3.

2.4.11.1 Performance evaluation metrics

We follow the performance evaluation metrics used in [11] and report three numbers for each setting: Area under the curve (AUC), Equal Error Rate (EER) and Accuracy. AUC measures the area under ROC curves and lies in the range 0 to 1 where higher value corresponds to better performance. EER is the point where the false accept rate is equal to false reject rate. It also lies in the 0 to 1 with lower values indicating better performance. We use an optimal threshold to classify all pairs and calculate the classification accuracy. For the optimal threshold, we choose the value that provides highest classification accuracy on the cross validation set.

2.4.11.2 Results for frontal-to-frontal and frontal-to-profile protocols

The experimental results for frontal-to-frontal and frontal-to-profile protocols are summarized in Table 2.4. CNN-1 to CNN-4 results are obtained by using the same models and processing steps for IJB-B and IJB-C experiments. For the fusion part, since all detection scores for the images in CFP dataset are near 1, we simply average the similarity score for CNN-1 through CNN-4. Deep features and human results are directly cited from [11]. The performance is reported by averaging over

ten splits.

For the frontal-to-frontal setting, CNN-1 to CNN-4 all outperform both the deep features method and human performance in [11]. CNN-2_S and CNN-2_L perform almost identically. CNN-2 and CNN-3 perform similarly and their performances are slightly better than CNN-1 and CNN-4. Since performances of CNN-2 and CNN-3 have already saturated, fusion results for the five networks do not change much compared to CNN-2 or CNN-3. For the frontal-to-profile setting, different algorithms begin to show significant difference in performance. CNN-1 results are slightly worse than human performance but are 2% better than CNN-4. On the other hand, CNN-2 and CNN-3 both surpass human performance by more than 2%. Another interesting finding is that while the performance for different algorithms do not vary much in frontal-to-frontal protocol, the performance drops from frontal-to-frontal to frontal-to-profile is quite different among the compared algorithms. Generally speaking, better algorithms are more robust to extreme yaw variations and always have smaller performance degradation for frontal-to-profile setting. In particular, CNN-2_S has the smallest performance drop of 1.6% from frontal-to-frontal to frontal-to-profile, which is similar to human performance. However, if we compare the results with Section 2.4.3, even the best results are still severely affected by pose variations. This is because the IJB-B and IJB-C datasets contain other challenging factors and pose variations can still degrade performance once combined with these factors. Therefore, even for state-of-the-art face models, there is still room to improve robustness to extreme pose variations.

2.5 Conclusions

In this chapter, we present the results of comprehensive experiments performed to study the effects of covariates on unconstrained face verification performance. Our evaluations are based on deep learning networks and large training data sets. We also curate the training data by exploiting gender information and achieve improved performance. Experimental results on the overall protocols of IJB-B and IJB-C covariate verification tasks show the outstanding performance of five implemented deep models and their score-level fusion. However, when we focus on each specific covariate, we find that many covariates still significantly affect the verification performance. Pose variations and occlusions are the top confounding factors that could cause performance drop by large margins. Indoor performance is much better than outdoors. On the other hand, the difficulty of unconstrained face verification varies significantly for different demographic groups. Age, gender and skin tone impact performance. Specifically, males are easier to verify than females and old subjects generally performs better than young ones. For skin tone, light pink achieves the best performance while medium-dark brown performs the worst. However, since IJB-B and IJB-C show very different tendencies on skin tone groups, we are not be able to draw a clear conclusion on its effects.

Most of the findings discussed above confirm the conclusions of previous studies. However, there are also some new findings that were rarely mentioned by other studies or somewhat surprising. First, we find that verification performance does not increase monotonically as subjects get older. In contrast, performance begins

to drop for age group of $[50, 65]$ and 65+. This result is different from most studies which claim older subjects are always easier to be recognized. However, since most of other studies did not have a sufficient number of older subjects to analyze, their results still make sense because middle age group performs better than children and teenagers. Second, we observe that extreme roll angle differences between faces still affect performance substantially. This result is unexpected as roll variations should be eliminated by face alignment. Therefore, we conclude that face alignment performance needs to get better when faces are in extreme roll angles.

Finally, we investigate the mixed effects of multiple covariates. First, males and females show very different trends on the effects of age groups. For males, performance first increases then drops when age goes up while for females, older age groups always perform better. On the other hand, the interaction between gender and skin tone does not show clear trends. Second, when we consider indoor/outdoor and occlusion together, we find that indoor and nose-mouth visibility must be satisfied simultaneously to achieve good performance. However, indoor/outdoor and yaw angle difference can affect the performance independently.

Chapter 3: Pose-Robust Face Verification by Exploiting Competing Tasks

3.1 Overview

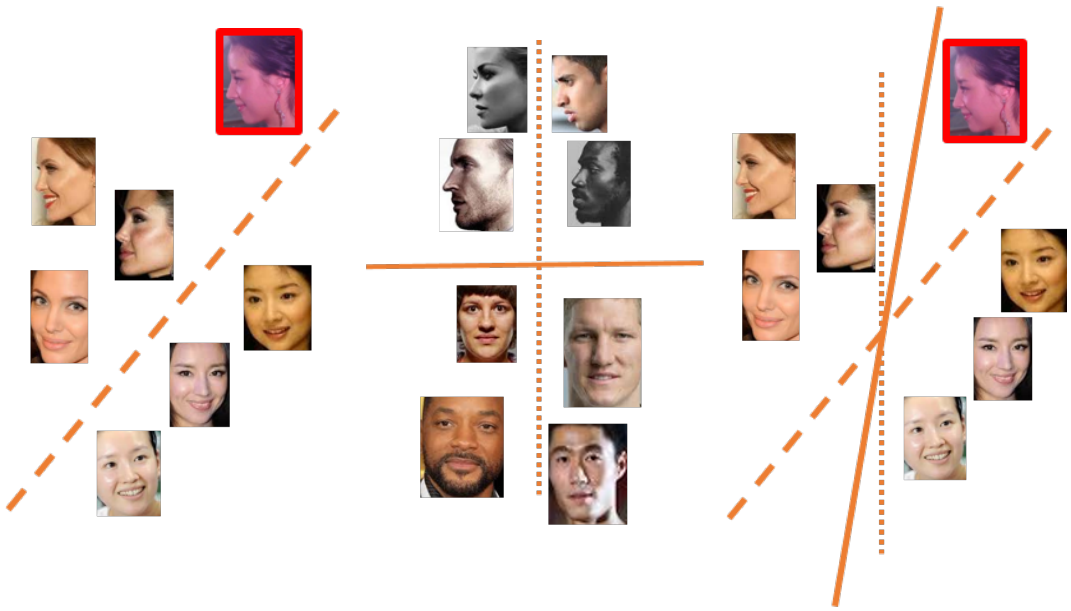
In the unconstrained face verification problem, pose variation is one of the most difficult factor to handle as face images from various poses lie in a highly nonlinear manifold, where the structure can hardly be captured and modeled [56]. In addition, possible pose variations in the training and test set may introduce large domain mismatch. Therefore, pose-invariant face verification has attracted significant attention [17]. Some previous works seek to learn a pose-invariant representation [19, 57], while others focus on multi-view common subspace learning [20], or synthesize faces based on generic 3D models [21]. In this chapter, we propose a pose-robust metric learning framework for face verification by cooperating with the pose verification task. Based on the intuition that the metrics for the two tasks are competing with each other, we jointly learn the projection matrices for the two tasks and add an orthogonal regularization constraint. The orthogonal regularization enforces the metrics for the two tasks to be uncorrelated with each other and to capture different kinds of information in the features. Therefore, the learned

metrics for the main task extracts pose-robust identity information and discounts the pose-sensitive information contained in the metrics for the auxiliary task.

To better understand why the two tasks are competing, we give a simple example of face identification and pose classification, which are closely related to the task of face and pose verification. In Fig. 3.1(a), an identity classifier for face identification is trained to classify two different subjects. However, some of the training data is biased, *e.g.*, for some particular persons, the number of training samples is limited and most of the faces are frontal or near-frontal. In this situation, given a new profile face of the person, it is very likely that the face will be classified as someone else who has plenty of profile faces in the training set. To solve this problem, we can exploit the information from pose classification. As illustrated in Fig. 3.1(b), the pose classifier indicates the most pose-sensitive orientation, while the normal vector of the pose classifier represents the least discriminative direction for poses (the dashed line). This observation suggests that the normal vector of the pose classifier can provide helpful information for the identity classifier to achieve pose robustness, which is shown in Fig. 3.1(c). In other words, adding an orthogonal constraint between the classifiers for the two tasks would make the identity classifier more pose-robust.

3.2 Proposed Approach

In this section, we describe the proposed metric learning framework. After the metrics is learned, we demonstrate how to use them for pose-invariant face



(a) (b) (c)

Figure 3.1: Training a face recognition classifier by coordinating with pose information. (a) a face classifier trained with only identity information. The red boxed face is wrongly classified due to the bias in the training data. (b) a pose classifier trained using pose labels, and the classifier (solid line) is discriminative only with respect to poses. (c) using the normal direction of the pose classifier (vertical dashed line) to regularize the face classifier. The red boxed face is correctly classified by the new classifier (solid line) after regularization.

verification. For the following subsections, we first briefly review the joint Bayesian metric learning as the baseline method and then present the the details of the proposed algorithm.

3.2.1 Joint Bayesian Metric Learning

The joint Bayesian method is widely used for face verification tasks [58, 59]. The main idea behind the joint Bayesian method is to model the joint distribution of a pair of feature vectors and maximize the log likelihood ratio of intra-class and inter-class distributions [58]. The final formulation of joint Bayesian can also be interpreted as a combination of Mahalanobis distance and projected cosine similarity. Instead of using statistical techniques to generate the solution, Chen *et al.* [59] directly optimized the distance in a large-margin framework as follows:

$$\underset{\mathbf{W},\mathbf{V},b}{\operatorname{argmin}} \sum_{ij} \max\{0, \alpha - l_{ij}(b - d_{\mathbf{W}}(\mathbf{x}_i, \mathbf{x}_j) + 2s_{\mathbf{V}}(\mathbf{x}_i, \mathbf{x}_j))\} \tag{3.1}$$

where $d_{\mathbf{W}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{W}^T \mathbf{W}(\mathbf{x}_i - \mathbf{x}_j)$ is the Mahalanobis distance and $s_{\mathbf{V}}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{V}^T \mathbf{V} \mathbf{x}_j$ is the projected similarity. Both $\mathbf{W} \in \mathbb{R}^{n \times d}$ and $\mathbf{V} \in \mathbb{R}^{n \times d}$ are the projection matrices. Here the projection matrices are either low rank embeddings ($n < d$) or full rank transformations ($n = d$). $l_{ij} = 1$ if $\{\mathbf{x}_i, \mathbf{x}_j\}$ is a positive pair and $l_{ij} = -1$, otherwise. b is the bias and α is the margin parameter. The optimization problem in (3.1) can be efficiently solved by Stochastic Gradient Descent (SGD) method. The details can be found in [59].

3.2.2 Learning by Exploiting Competing Tasks

In order to fully exploit pose-sensitive information and coordinate with face verification task, we construct an auxiliary competing task called pose verification. Different from the main task of face verification, pose verification aims to learn the pose-sensitive information in features. More specifically, given a pairs of features $\{\mathbf{y}_i, \mathbf{y}_j\}$, the algorithm generates large (small) similarity scores when $\mathbf{y}_i$ and $\mathbf{y_j}$ have similar (different) poses. One key property is that the similarity scores should only depend on the similarity of poses, regardless of whether the features come from the same person or not.

For the main task of face verification, we rewrite the hinge-loss objective function in (3.1) as $\mathcal{L}_f(\mathbf{W_f}, \mathbf{V_f}, b_f)$. Similarly, we can denote the objective function for pose verification as $\mathcal{L}_p(\mathbf{W_p}, \mathbf{V_p}, b_p)$. Intuitively, the competing relationships between the main task and the auxiliary task suggest that the projections for face verification and that for pose verification should be uncorrelated. In addition, the features used for both tasks should be extracted from the same feature pool, which makes the projection matrices for different tasks comparable. The joint multi-task model is formulated as:

$$\underset{\substack{\mathbf{W}_f,\mathbf{V}_f,b_f,\\ \mathbf{W}_p,\mathbf{V}_p,b_p}}{\operatorname{argmin}} \mathcal{L}_f(\mathbf{W}_f, \mathbf{V}_f, b_f) + \mathcal{L}_p(\mathbf{W}_p, \mathbf{V}_p, b_p) + \lambda_1 \|\mathbf{W}_f^T \mathbf{W}_p\|_F^2 + \lambda_2 \|\mathbf{V}_f^T \mathbf{V}_p\|_F^2 \quad (3.2)$$

where λ_1, λ_2 are regularization parameters. The projection matrices are chosen to be low-rank embeddings and can be initialized by applying principal component

analysis (PCA) to the training data. The low rank embeddings not only efficiently simplify the computational complexity, but also eliminate the underlying noise and provide imrpoved performance [45]. Although the optimization of the projection matrices is a non-convex problem, the algorithm still yields good results [60].

The objective function in (3.2) has two parts. The first two terms jointly minimize the verification errors for both tasks, while the last two terms enforce the orthogonal regularizations on the projection matrices for face and pose verification. Compared to the baseline method, the projection matrices for face verification learned by the proposed framework are more robust to pose variations because they not only encode the identity-sensitive information, but also mitigate the pose-sensitive information by coordinating with the pose verification task.

We use SGD to optimize the objective function in (3.2). In each iteration, we randomly pick up a positive or negative pair of training samples $\{\mathbf{x}_i, \mathbf{x}_j\}$ for face verification and $\{\mathbf{y}_i, \mathbf{y}_j\}$ for pose verification. If the similarity condition is violated, we update $\mathbf{W}_f, \mathbf{W}_p, \mathbf{V}_f, \mathbf{V}_p, b_f, b_p$ as follows:

$$
\begin{aligned}
\mathbf{W}_f^{t+1} &= \begin{cases} \mathbf{W}_f^t, & \text{if } l_{ij}\rho_{ij} \geq \alpha_f \\ \mathbf{W}_f^t - \tau_f(l_{ij}\mathbf{W}_f^t\mathbf{\Psi}_{ij} + \lambda_1\mathbf{W}_p^t\mathbf{W}_p^{tT}\mathbf{W}_f^t), & \text{otherwise,} \end{cases} \\
\mathbf{V}_f^{t+1} &= \begin{cases} \mathbf{V}_f^t, & \text{if } l_{ij}\rho_{ij} \geq \alpha_f \\ \mathbf{V}_f^t + \tau_f(l_{ij}\mathbf{V}_f^t\mathbf{\Gamma}_{ij} + \lambda_2\mathbf{V}_p^t\mathbf{V}_p^{tT}\mathbf{V}_f^t), & \text{otherwise,} \end{cases} \\
b_f^{t+1} &= \begin{cases} b_f^t, & \text{if } l_{ij}\rho_{ij} \geq \alpha_f \\ b_f^t + \tau_f l_{ij}, & \text{otherwise,} \end{cases}
\end{aligned}
\tag{3.3}
$$

$$
\begin{aligned}
\mathbf{W}_p^{t+1} &= \begin{cases} \mathbf{W}_p^t, & \text{if } a_{ij}\theta_{ij} \geq \alpha_p \\ \mathbf{W}_p^t - \tau_p(a_{ij}\mathbf{W}_p^t\mathbf{\Phi}_{ij} + \lambda_1\mathbf{W}_f^{t+1}\mathbf{W}_f^{t+1^T}\mathbf{W}_p^t), & \text{otherwise,} \end{cases} \\
\mathbf{V}_p^{t+1} &= \begin{cases} \mathbf{V}_p^t, & \text{if } a_{ij}\theta_{ij} \geq \alpha_p \\ \mathbf{V}_p^t + \tau_p(a_{ij}\mathbf{V}_p^t\mathbf{\Delta}_{ij} + \lambda_2\mathbf{V}_f^{t+1}\mathbf{V}_f^{t+1^T}\mathbf{V}_p^t), & \text{otherwise,} \end{cases} \\
b_p^{t+1} &= \begin{cases} b_p^t, & \text{if } a_{ij}\theta_{ij} \geq \alpha_p \\ b_p^t + \tau_p a_{ij}, & \text{otherwise,} \end{cases}
\end{aligned} \tag{3.4}
$$

where τ_f, τ_p are the learning rates, l_{ij}, a_{ij} are training labels, $\mathbf{\Psi}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T$, $\mathbf{\Phi}_{ij} = (\mathbf{y}_i - \mathbf{y}_j)(\mathbf{y}_i - \mathbf{y}_j)^T$, $\mathbf{\Gamma}_{ij} = \mathbf{x}_i\mathbf{x}_j^T + \mathbf{x}_j\mathbf{x}_i^T$, $\mathbf{\Delta}_{ij} = \mathbf{y_i}\mathbf{y}_j^T + \mathbf{y}_j\mathbf{y}_i^T$, $\rho_{ij} = b_f - d_{\mathbf{W}_f}(\mathbf{x}_i, \mathbf{x}_j) + 2s_{\mathbf{V}_f}(\mathbf{x}_i, \mathbf{x}_j)$, $\theta_{ij} = b_p - d_{\mathbf{W}_p}(\mathbf{y}_i, \mathbf{y}_j) + 2s_{\mathbf{V}_p}(\mathbf{y}_i, \mathbf{y}_j)$. Instead of updating at every iteration, the regularization terms are updated only when the similarity condition is violated. In practice, this strategy significantly reduces the computational complexity but yields similar results.

Although the same deep features are used for both tasks, the difficulties for the main task (face verification) and the auxiliary task (pose verification) are very different since the deep neural networks are trained solely by the identity labels. Therefore, the features are more specific to identity information. To solve this problem, we pre-train the pose verification model using the pose labels. The pre-trained model can thus encode more pose information from the features and provide a good initialization of the pose metrics for multi-task learning. The whole procedure is summarized in Algorithm 3.

Algorithm 1 Stochastic Gradient Descent for Multi-Task Metric Learning

Input: Training pairs X with associated labels L for face verification and pairs Y with labels A for pose verification, margin α, parameter λ_1, λ_2, maximum iteration number N

1: **Pre-train Pose Model**: Pre-train the pose model $\mathbf{W}_{p0}, \mathbf{V}_{p0}, b_{p0}$ using (3.1)

2: **Initialization**: Initialize $\mathbf{W}_{f0}, \mathbf{W}_{f0}$ using PCA, $b_{f0} = 0$, $\mathbf{W}_{p0}, \mathbf{V}_{p0}, b_{p0}$ from the pre-trained model

3: **for** t = 1:N **do**

4: Randomly pick up a pair $\{\mathbf{x}_i, \mathbf{x}_j\}$, update the face verification model $\mathbf{W}_f^t, \mathbf{V}_f^t, b_f^t$ using (3.3)

5: Randomly pick up a pair $\{\mathbf{y}_i, \mathbf{y}_j\}$, update the pose verification model $\mathbf{W}_p^t, \mathbf{V}_p^t, b_p^t$ using (3.4)

6: **end for**

Output: Projection matrices $\mathbf{W}_f, \mathbf{W}_p, \mathbf{V}_f, \mathbf{V}_p$ and biases b_f, b_p

3.2.3 Pose-Robust Face Verification

Although the joint model learns two metrics, one for the main task and the other for the auxiliary task, we only utilize the face verification model to achieve improved performance on the main task. Once the projection matrices $\mathbf{W}_f, \mathbf{V}_f$ are learned, we calculate the similarity scores of the test pairs $\{\mathbf{x}_i, \mathbf{x}_j\}$ as

$$sim(\mathbf{x}_i, \mathbf{x}_j) = 2s_{\mathbf{V}_f}(\mathbf{x}_i, \mathbf{x}_j) - d_{\mathbf{W}_f}(\mathbf{x}_i, \mathbf{x}_j) \tag{3.5}$$

The learned bias b_f is not included in the final formulation of the similarity scores because the bias is only an uniform offset and will not change the final performance.

3.3 Experiments

In this section, we evaluate the proposed approach on three challenging datasets: IARPA Janus Benchmark A (IJB-A), Janus Challenging Set 3 Covariates (CS3_cov), and Celebrities in Frontal-Profile (CFP). We begin with introducing the details of the datasets and the experimental settings. Then discussions on the experimental results are presented.

IARPA Janus Benchmark A [61]: This dataset contains 500 subjects with a total of 5,397 images and 2,042 videos. For the evaluation purpose, ten splits are generated based on different training / test set division. Each training set and test set contains 333 subjects and 167 subjects, respectively. The dataset contains many extreme pose and illumination variations and some sample images are shown

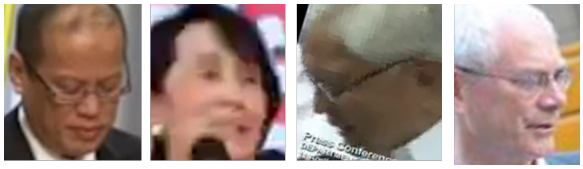


Figure 3.2: Sample images in IJB-A dataset.

in Figure 3.2. The IJB-A verification protocol has around 11,748 pairs of gallery-to-probe templates (1,756 positive and 9,992 negative pairs), with each templates containing a mixture of images and video frames.

Janus Challenging Set 3 Covariates: The Janus Challenging Set 3 (CS3) dataset contains 1871 subjects and 68716 images and video frames. The covariates protocol aims to focus on the effect of eight different covariates (age, eyes visible, facial hair, forehead visible, gender, indoor, nose and mouth visible and skin tone) on the verification performance. The protocol evaluates 20,866,895 pair of templates (5,961,839 positive and 14,905,056 negative pairs) where each template contains one image or frame. Some sample images are shown in Figure 3.3.

Celebrities in Frontal-Profile [11]: This dataset investigates the influence of extreme pose variations on the face verification performance. The dataset contains 500 subjects, with 10 frontal and 4 profile images for each subject. Most of the profile images are in extreme poses and some sample images are shown in 3.4. For the evaluation protocol, there are two settings: frontal-to-frontal and frontal-to-profile face verification. For each setting, it consists of ten disjoint splits and each split has 350 positive and 350 negative pairs. The final performance is averaged over ten splits. In this chapter, we focus on pose-variant face verification and thus only

run the experiments for the frontal-to-profile protocol.

3.3.1 Experimental Setup

Features: The deep CNN features used in all the experiments of this work are extracted using the architecture proposed in [59]. The model consists of ten convolutional layers, five pooling layers and one fully connected layer and is trained using the CASIA-WebFace dataset [62]. The output of the pool5 layer is used as the final features and the dimensionality of the features is 320. All the features are l_2 normalized before computing the similarity score. In IJB-A dataset, there are more than one samples in each templates. We use the media averaging strategy which is similar to the one reported in [45].

Auxiliary Task Design: The face poses used in this chapter are estimated using the approach discussed in [63]. Since the estimated poses may not be perfectly accurate, we cluster the poses into groups and treat the poses equally within each group. For CS3 and IJB-A datasets, we divide the poses into four groups and for the CFP dataset three groups are generated. In order to avoid the identity bias in the pose groups (some subjects may have more large poses than others), we randomly choose samples from different subjects for each pose group. The positive pairs are selected by randomly picking up two samples in the same group and the negative pairs consist of samples picked from different groups.

Accuracy Metrics: To evaluate the CS3 Covariates and IJB-A verification performance, we follow the evaluation protocol defined in [61]. The original protocol

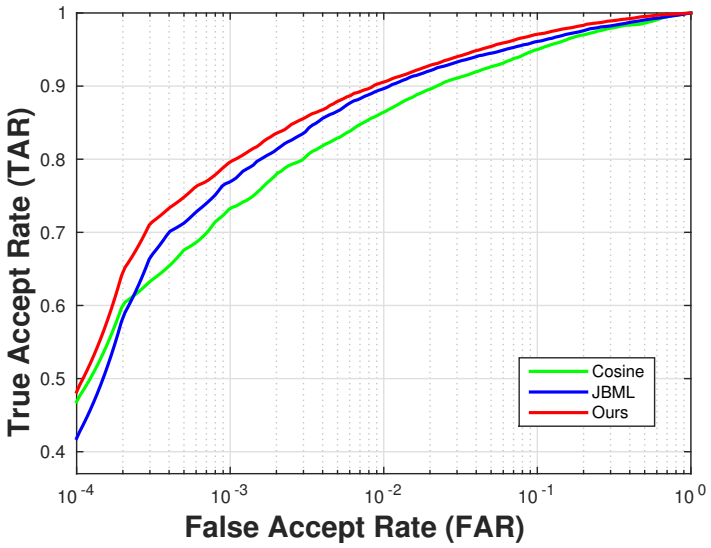
Figure 3.3: Sample images in CS3 dataset.

Figure 3.4: Sample images in CFP dataset.

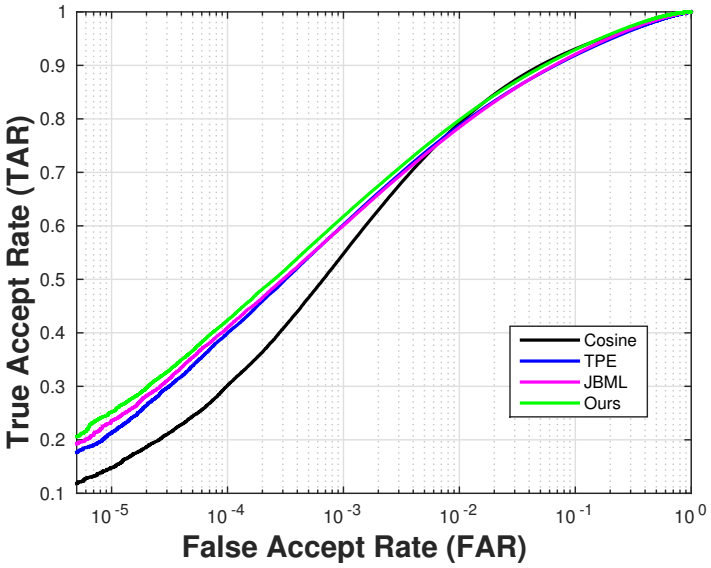
reports ROC curves as well as the True Acceptance Rate (TAR) when False Alarm Rate (FAR) equals $10^{-3}, 10^{-2}, 10^{-1}$. For the CS3 Covariates protocol, the total number of pairs is extremely large (about 20 million pairs), thus we also report the TAR at FAR= $10^{-5}, 10^{-4}$. In addition, we also analyze the performance under different covariates that are related with poses (eyes visibility, forehead visibility). The accuracy metrics used for the CFP dataset follow the protocol in [11]. AUC

Method	TAR@FAR = 10^{-3}	TAR@FAR = 10^{-2}	TAR@FAR = 10^{-1}
Cosine	0.734±0.042	0.864±0.014	0.950±0.006
JBML [59]	0.799±0.022	0.906±0.010	0.973±0.004
TPE [45]	0.813±0.020	0.900±0.010	0.964±0.005
Proposed Method	**0.814±0.027**	**0.913±0.010**	**0.977±0.003**

Table 3.1: Verification results for the IJB-A dataset. Results are averaged over ten splits.



(a) ROC curves for IJB-A



(b) ROC curves for CS3 Covariates

Figure 3.5: ROC curves for the CS3 Covariates and the IJB-A dataset. The results are averaged over 10 splits for the IJB-A dataset.

and EER are computed for each split, as well as the classification accuracy. The performance is reported by averaging over ten splits. For classification accuracy, we select the threshold that provides highest accuracy on the training set.

Parameters: We set the margin $\alpha_f = \alpha_p = 0.001$. Intuitively, a small margin encourages the projection matrices to be updated only by the hard negative/positive pairs since small margins result in less strict condition than large margins. The hard negative mining yields a similar idea and has been widely used for SGD updating. Based on the above observation, we choose the margin to be a small value. The initialization of the projection matrices for the CS3 dataset is the whitening PCA of the training data while for IJB-A and CFP datasets, we find that initialization using WCPA makes the projection matrices have very large values and thus they become unstable. Therefore, we use PCA to initialize the projection matrices. The learning rates are set to be $3 \times 10^{-4}, 5 \times 10^{-3}, 3 \times 10^{-3}$ for CS3, IJB-A and CFP respectively.

Method	TAR@FAR = 10^{-5}	TAR@FAR = 10^{-4}	TAR@FAR = 10^{-3}	TAR@FAR = 10^{-2}	TAR@FAR = 10^{-1}
Cosine	0.148	0.302	0.548	0.792	**0.931**
JBML [59]	0.236	0.410	0.601	0.784	0.921
TPE [45]	0.213	0.400	0.602	0.788	0.919
Proposed Method	**0.252**	**0.424**	**0.618**	**0.798**	0.930

Table 3.2: Verification results for the CS3 covariates protocol.

3.3.2 Evaluation Results for the IJB-A dataset

Compared Methods: The experimental results of the proposed approach are compared with two baseline methods, the cosine similarity and the joint Bayesian metric learning (JBML). The cosine similarity measure is computed directly from the raw features while JBML is learned by using the identity labels of the training data. In addition, we also compare with the triplet probabilistic embedding (TPE) method [45]. We use the same features to compute the similarity scores for cosine similarity, JBML and our method. In contrast, the results for TPE are directly cited from [45] and we observe that the raw features used in [45] have a better baseline performance than our features.

Table 3.1 summarizes the results for the IJB-A dataset. It can be seen that the proposed metric learning method outperforms the cosine similarity, JBML baselines and TPE method at all the FARs. In addition, considering the fact that TPE has better features, the proposed method achieves competitive performance. To better visualize the performance, the ROC curves are shown in Fig. 3.5(a).

3.3.3 Evaluation Results on CS3 Covariates

General Performance: For a fair comparison, the same features are used for all the methods. We plot the ROC curves for the CS3 Covariates protocol in Fig. 3.5(b) and Table 3.2 shows the True Acceptance Rate (TAR) versus False Alarm Rate (FAR) at different values. We notice that the proposed approach consistently improves the JBML baseline and outperforms TPE at all FARs. Interestingly, we

Eye Visible	Method	TAR@FAR = 10^{-5}	TAR@FAR = 10^{-4}	TAR@FAR = 10^{-3}	TAR@FAR = 10^{-2}	TAR@FAR = 10^{-1}
Same	Cosine	0.146	0.297	0.546	0.798	**0.935**
	JBML [59]	0.219	0.404	0.605	0.792	0.926
	Proposed Method	**0.243**	**0.418**	**0.622**	**0.805**	0.933
Different	Cosine	0.118	0.254	0.468	0.705	**0.888**
	JBML [59]	0.220	0.344	0.503	0.694	0.874
	Proposed Method	**0.221**	**0.350**	**0.515**	**0.709**	0.886

Table 3.3: Covariates analysis on eye visibility. *Same* represents that the two face images in a pair are both eye visible or non-visible, and *Different* means that one of the faces is eye visible while the other is non-visible.

Forehead Visible	Method	TAR@FAR = 10^{-5}	TAR@FAR = 10^{-4}	TAR@FAR = 10^{-3}	TAR@FAR = 10^{-2}	TAR@FAR = 10^{-1}
Same	Cosine	0.145	0.305	0.559	0.796	**0.930**
	JBML [59]	0.219	0.413	0.609	0.788	0.919
	Proposed Method	**0.245**	**0.430**	**0.624**	**0.798**	0.926
Different	Cosine	0.161	0.294	0.530	0.785	0.933
	JBML [59]	0.260	0.404	0.586	0.777	0.923
	Proposed Method	**0.267**	**0.415**	**0.608**	**0.797**	**0.935**

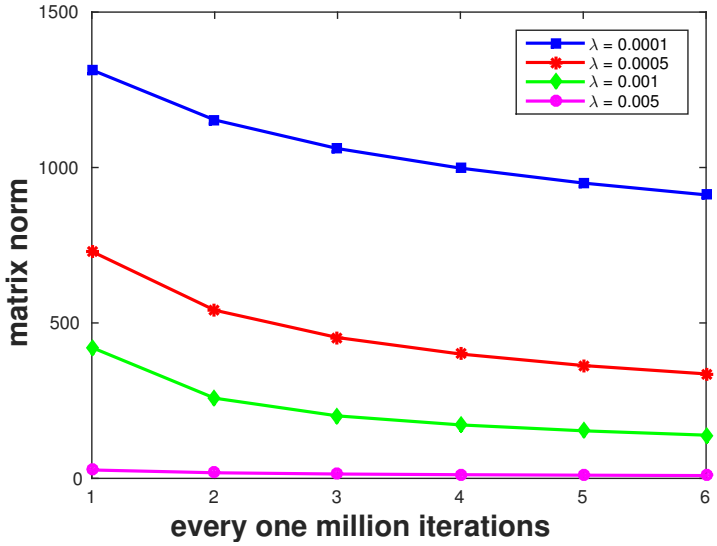
Table 3.4: Covariates analysis on forehead visibility. *Same* represents that the two face images in a pair are both forehead visible or non-visible, and *Different* means that one of the faces is forehead visible while the other is non-visible.

Method	Accuracy	EER	AUC
Cosine	0.904	0.094	0.967
Sengupta *et al.* [11]	0.849	0.150	0.930
JBML [59]	0.924	**0.068**	**0.981**
Proposed Method	**0.929**	0.071	**0.981**

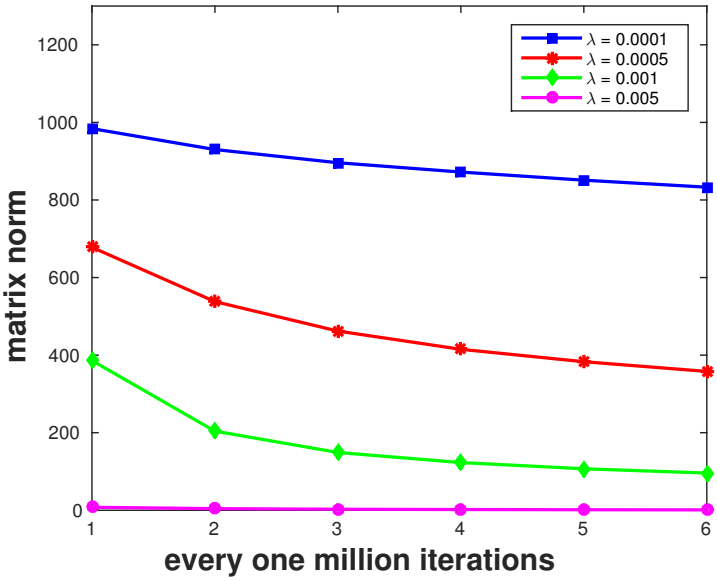
Table 3.5: Verification results for the frontal-to-profile protocol for the CFP dataset. Results are averaged over ten splits.

observe that JBML and TPE perform slightly worse than the cosine baseline at FAR= $10^{-2}, 10^{-1}$. Possibly, this is because the training set may not contain sufficient face images with large poses and the learned metrics become biased to frontal or near-frontal faces. When the projection matrices are applied to the test data, where many faces are in extreme poses, the performance goes down. In contrast, the proposed method explicitly avoids the pose informations in the metrics for the main task, and thus it is more pose-robust than the baseline metric.

Covariates Analysis: In order to better understand how the covariates affect the verification performance, we evaluate two pose-relevant covariates, eye visibility and forehead visibility, and present the results. Tables 3.3 and 3.4 show the experimental results for cosine, JBML and the proposed method over eye and forehead visibility. *same* represents the same visibility (both visible or non-visible) and *different* means different visibility for the compared faces. Generally, the performance for same visibility is better than that for different visibility. Since eye and forehead



(a)



(b)

Figure 3.6: The Frobenius norm of the regularization terms for W and V matrices over iterations for CS3 dataset.

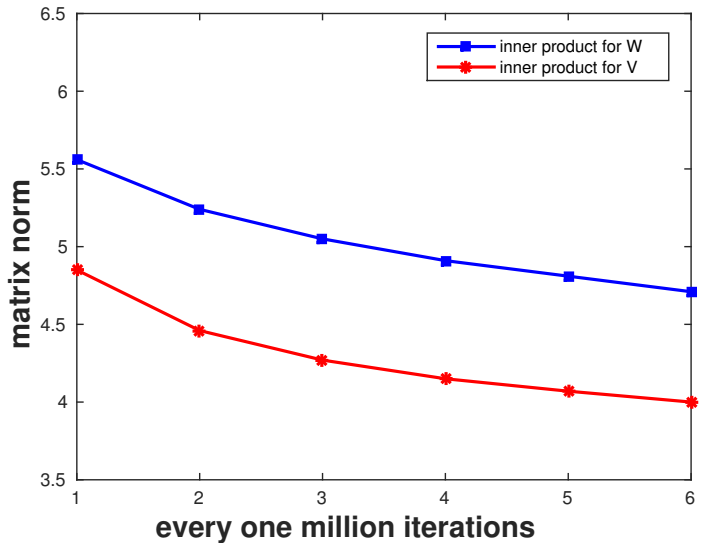


Figure 3.7: The Frobenius norm of the regularization terms for W and V matrices over iterations for CFP dataset.

visibility partially reflect the pose variations, it demonstrates that pose variations indeed degrade the performance. We notice that the proposed approach consistently outperforms the joint Bayesian baseline and cosine similarity for most cases except slightly worse than cosine similarity at FAR= 10^{-1}. Moreover, the improvement of our method over the baseline shows similar trends for *same* and *different* visibility cases. This reveals that the pose variations still exist, though smaller than that in different visibility case, in the same visibility situation.

Regularization Parameter Analysis: The regularization parameter λ controls the orthogonality of the projection matrices for the two tasks. We investigate the function of the regularization terms by varying the values of λ. The Frobenius norm of the inner product of the projection matrices for the two tasks are shown in Fig 3.6. We can see that the Frobenius norm monotonically goes down as the itera-

tions increase and a larger λ results in a more strict regularization on the projection matrices. When λ is large enough (typically larger than 0.005), the Frobenius norm becomes small and does not change much. We also run experiments to see how the performance changes with different λ's and do not notice much difference when λ changes from 10^{-4} to 5×10^{-3}.

3.3.4 Evaluation Results on CFP dataset

For the CFP dataset, the experimental results are given in Table 3.5. Surprisingly, we see that the proposed approach only slightly outperforms the JBML baseline on accuracy for 0.5% and performs a bit worse on EER. Intuitively, the learned metrics should alleviate the pose mismatch in the test pairs and improve the JBML performance. We further conduct experiments to see the underlying reasons for this issue. We find that the pose metric converges much faster than the identity metric. The accuracy for the pose verification is almost 100%. Considering the fact that the dataset only consists frontal and profile faces, the learned pose metric is not discriminative enough to small pose differences. At the same time, we draw the plot in Fig. 3.7 that the regularization term does not change much during joint training. This further demonstrates that the regularization term does not affect the face metric much.

3.4 Conclusion

In this chapter, we showed the benefit of cooperating with the pose verification task for pose-robust face verification. We proposed a joint model to learn the metrics for the two tasks together and enforced an orthogonal regularization on the learned projection matrices for the two tasks. By excluding the information contained in the auxiliary task, the learned metric for face verification is more pose-robust. We conducted extensive experiments on three challenging datasets and the experimental results show that the proposed approach improves the baseline methods and is competitive with the state-of-the-art.

Chapter 4: Incremental Dictionary Learning for Unsupervised Domain Adaptation

4.1 Overview

Classification tasks often assume that training and test data are drawn from the same distribution. However, this assumption is often challenged by real applications. For example, face recognition models trained on frontal faces with good resolution may be called upon to classify non-frontal or blurred faces. This domain shift has resulted in a large drop in classification performance and many domain adaptation (DA) methods have been developed to address this problem [64–68]. There are two main settings for DA: semi-supervised DA allows a few class labels in the target domain and in the case of unsupervised DA, target labels are not available. In this chapter, we focus on the more difficult unsupervised setting.

One class of unsupervised methods learns a transformation and project samples from both domains into a common subspace, in which the distribution divergence between the two domain becomes smaller [65–68]. Others attempt to reduce the domain mismatch by reweighting or selecting some source samples [69, 70]. In contrast, some bootstrapping-based DA methods [64,71–73] use the source classifier

to predict some target labels and then add them to the source domain to adapt the initial classifier.

In this chapter, we propose an incremental dictionary learning-based method which explicitly reduces the cross-domain divergence, and simultaneously performs adaptation and classification. Specifically, we iteratively find some *supportive samples* in the target domain and add them to the source domain. These supportive samples have two nice properties. First, the predicted labels of the supportive samples are reliable. So they are used to train a more powerful classification model. Second, the supportive samples are close to the source domain. So they reduce the domain dissimilarity. In addition, a good stopping criterion is crucial for efficient adaptation. We introduce a domain similarity measure and only conduct adaptation when the domain similarity value increases after each iteration. In this way, we automatically guarantee that our adaptation will monotonically reduce the domain mismatch.

4.2 Proposed Approach

In this section, we first present the proposed incremental dictionary learning-based DA method. We will then introduce a domain similarity measure and give some theoretical analysis to prove the effectiveness of the proposed method. We begin with describing some notations used in the chapter.

We use $X^s = X^{(0)} = \{x_i^s\} \in R^{d\times N_s}$, $X^t = \{x_i^t\} \in R^{d\times N_t}$ to denote the data from source and target domains, where N_s , N_t denote the number of samples

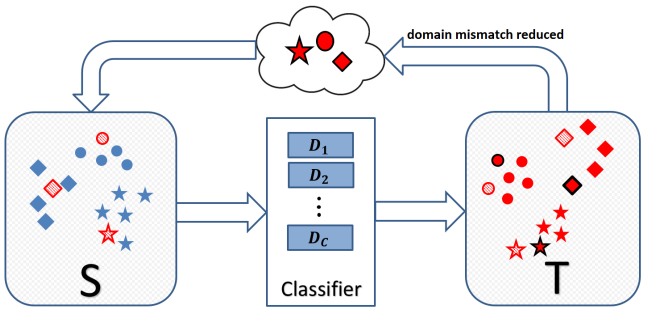


Figure 4.1: Scheme of the incremental dictionary learning for domain adaptation. The original source data is colored in *blue* and the target data is colored in *red*. Different shapes represent different classes. The red samples with shadow indicate the previously selected supportive samples that have been added to the source domain. The red samples with black border represent the supportive samples selected in the current iteration.

respectively, and d is the dimension of data. Let $L = \{1, .., C\}$ represent the existing label set. Let $D^{(0)} = [D_1^{(0)}|...|D_C^{(0)}]$ denote the original dictionary trained on source domain where $D_j^{(0)} \in R^{d \times K}$ denote the sub-dictionary that corresponds to class j, and K represents the number of atoms in each class specific sub-dictionary. Let $P \in R^{N_t \times C}$ denote the confidence matrix with each element $p_{ij} \in [0, 1]$ representing the probability that target sample x_i^t belongs to the class j. Let $W \in R^{N_t \times C}$ denote the binary selection matrix with each element $w_{ij} \in \{0, 1\}$ indicating whether the target sample x_i^t is selected as supportive samples for class j. $X^{(k)}$, $D^{(k)}$, $P^{(k)}$, $W^{(k)}$ denote the augmented source domain, dictionary, confidence and selecting matrix in the k^{th} iteration.

4.2.1 Incremental Dictionary Learning for DA

Given the dictionary $D^{(k)}$, we want to select a subset of target samples as supportive samples. We have two constraints on this selection. First, the supportive samples selected in the previous iterations should be excluded as we want to add new data for adaptation. Second, we select equal number of supportive samples for each class to ensure class balance during adaptation [69]. With these two constraints, we select the most confident samples that minimize the reconstruction error when represented by $D^{(k)}$. Then we update the augmented source domain by adding supportive samples and retrain the dictionary. The stopping criterion is then checked to see whether adding new supportive samples will reduce the domain dissimilarity. The proposed approach is shown in Fig. 4.1 for better understanding.

Confidence Matrix Update: In the $(k+1)^{th}$ iteration, We update the confidence matrix $P^{(k+1)}$ using the current class-specific dictionaries $D^{(k)} = [D_1^{(k)}|...|D_C^{(k)}]$:

$$p_{ij}^{(k+1)} = \begin{cases} \frac{\frac{1}{\sqrt{2\sigma^2}}\exp(-\frac{e_{ij}^{(k+1)}}{2\sigma^2})}{\sum_{l=1}^{C}\frac{1}{\sqrt{2\sigma^2}}\exp(-\frac{e_{il}^{(k+1)}}{2\sigma^2})} & \text{if} j = \underset{l}{\operatorname{argmax}}\, p_{il}^{(k+1)} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

where σ^2 is a normalization parameter and e_{ij} denotes the reconstruction error of target sample x_i^t using $D_j^{(k)}$:

$$e_{ij}^{(k+1)} = ||x_i^t - D_j^{(k)} \cdot \gamma_{ij}^{(k+1)}||_2^2 \tag{4.2}$$

where $\gamma_{ij}^{(k+1)}$ is the sparse code. Here $p_{ij}^{(k+1)} \neq 0$ only when j is the most likely class that sample i belongs to. This constraint guarantees that a sample cannot be selected as the supportive sample for multiple classes.

Supportive Samples Selection: We select new supportive samples using $W^{(k+1)}$ by solving the following optimization problem:

$$\begin{aligned} W_j^{(k+1)} &= \underset{W_j}{\operatorname{argmax}} \quad tr(W_j P_j^{(k+1)}) \\ s.t. \quad W_j^{(k+1)} &\cdot \sum_{l=1}^{k} W_j^{(l)} = 0, \quad ||W_j^{(k+1)}||_0 = Q, \quad j = 1, ..., C \end{aligned} \tag{4.3}$$

where $W_j \in R^{N_t \times N_t}$ are diagonal matrices with each element in the j^{th} column of W on the diagonal, *e.g.*, $W_j = diag\{w_{1j}, w_{2j}...\}$ and similarly $P_j = diag\{p_{1j}, p_{2j}...\}$. Q is the number of supportive samples for each class.

This objective function (4.3) maximizes the confidence of the selected supportive samples. The first constraint requires that the supportive samples in the $(k+1)^{th}$ iteration are disjoint from the previously chosen ones which ensures that we keep adding new supportive samples to the source domain. The second constraint ensures that the number of supportive samples for each class is balanced.

The solution to (4.3) is to find the corresponding Q supportive samples that maximize the confidence with the constraint that old supportive samples are excluded.

Augmented Source Domain Update: After selecting the supportive samples, we update the augmented source data by adding weighted supportive samples to current source data:

$$X_j^{(k+1)} = [X_j^{(k)} | X^t W_j^{(k+1)} P_j^{(k+1)}] \quad j = 1, ..., C \tag{4.4}$$

Since the labels of the supportive samples may have error, each selected supportive sample is weighted by its confidence. The weights indicate the reliability of the labels of the supportive samples and highly confident supportive samples will contribute more to the model.

Dictionary Update: Dictionary is updated by solving the following optimization problem:

$$D_j^{(k+1)} = \underset{D_j, Z_j}{\operatorname{argmin}} ||X_j^{(k+1)} - D_j \cdot Z_j||_F^2 + \lambda ||Z_j||_1 \quad j = 1, ..., C. \tag{4.5}$$

We solve (4.5) using the online dictionary learning method [74]. The dictionary obtained in the previous iteration is used as the initial dictionary in the next iteration. In this way, the computational cost is relatively low.

Stopping criterion: One trivial stopping criterion is to stop when there is no new supportive samples for one of the classes. But our goal is to guarantee that the adaptation monotonically reduces the domain shift. In this way, the classification error bound in target domain will decrease as stated in [75]. So we design a domain similarity measure and perform adaptation only when the domain similarity

increases after each iteration. The proposed approach is summarized in Algorithm 3.

4.2.2 Theoretical Analysis

In this section, we first introduce the domain similarity measure used for determining the stopping criterion. In order to quantify the domain similarity, several methods have been proposed [67, 76]. However, they need to design the dictionary or do PCA for both domains, which may be time consuming when data size is large. We introduce a simple domain similarity measure for X^s and X^t: $\rho(X^s, X^t) = \sqrt{\frac{1}{N_s N_t} \sum_i \sum_j (x_i^{sT} x_j^t)^2} = \sqrt{\frac{tr((X^s)^T X^t (X^t)^T X^s)}{N_s N_t}}$.

Since the classification accuracy on supportive samples is good, the main reason that causes the performance to drop in the target domain is that the source classifier behaves poorly on the non-supportive samples. It indicates that domain mismatch mainly lies between the source samples and the non-supportive samples. If the distance between supportive samples and non-supportive samples is smaller than the distance between the source domain and the non-supportive samples, selecting supportive samples can help reduce the domain mismatch and thus help classification as stated in [75]. The following theorem proves this notion and we present experimental results to validate the theoretical results in Section 4.3.

Theorem 1. *We divide the target samples into two part, supportive samples X_f and non-supportive samples X_n with N_f and N_n samples. With the definition of ρ above, and if $\rho(X_f, X_n) > \rho(X^s, X_n)$, then the domain similarity (or mismatch) will*

Algorithm 2 Incremental dictionary learning for unsupervised DA

Input: Initial dictionary $D^{(0)} = [D_1^{(0)}|...|D_C^{(0)}]$ learned from the source data, the target domain data X^t, similarity measure of two domains $\rho(X^s, X^t)$, number of supportive samples Q per class, parameters λ.

Output: Class labels for target data X^t.

k = 0

repeat

1. Confidence update: For each input data x_i^t, compute the reconstruction error on each $D_j^{(k)}$ using (4.2). Update each element of the confidence matrix $P^{(k+1)}$ using (4.1)

2. Supportive sample selection: For each class j, select the supportive samples using $W_j^{(k+1)}$ by maximizing (4.3).

3. Augmented source domain update: Update the augmented source domain $X_j^{(k+1)}$ by adding the new supportive samples:

$$X_j^{(k+1)} = [X_j^{(k)}|X^t W_j^{(k+1)} P_j^{(k)}] \quad j = 1, ..., C \tag{4.6}$$

4. Dictionary update: Update each class-specific dictionary $D_j^{(k+1)}$ by minimizing (4.5)

.

5. $k \leftarrow k + 1$.

until no supportive samples is selected or $\rho(X^{(k+1)}, X^t) <= \rho(X^{(k)}, X^t)$

classify X^t using the final dictionary.

increase(or decrease) when we add some supportive samples to the source domain:

$$\rho(X^s_{new}, X^t) > \rho(X^s_{old}, X^t) \tag{4.7}$$

where $X^s_{old} = X^s$ and $X^s_{new} = [X^s|X_f]$.

Proof. Since $\rho(X_f, X_n) > \rho(X^s, X_n)$, we have:

$$\begin{aligned}
&\rho^2(X_f, X_n) - \rho^2(X^s, X_n) \\
&= \frac{tr(X_n^T X_f X_f^T X_n)}{N_n N_f} - \frac{tr(X_n^T X^s X^{sT} X_n)}{N_n N_s} = \frac{tr((N_s X_f X_f^T - N_f X^s X^{sT}) X_n X_n^T)}{N_n N_s N_f} > 0.
\end{aligned}$$

Then:

$$\begin{aligned}
&\rho^2(X^s_{new}, X^t) - \rho^2(X^s_{old}, X^t) = tr({X^s_{new}}^T X^t X^{tT} X^s_{new}) - tr({X^s_{old}}^T X^t X^{tT} X^s_{old}) \\
&= \frac{tr(([X^s|X_f][\frac{X^{sT}}{X_f^T}][X_n|X_f][\frac{X_n^T}{X_f^T}])}{(N_s + N_f)(N_h + N_f)} - \frac{tr(([X_n|X_f][\frac{X_n^T}{X_f^T}] X^s X^{sT})}{N_s(N_h + N_f)} > 0 \\
&\Leftrightarrow \frac{tr((X^s X^{sT} + X_f X_f^T)(X_n X_n^T + X_f X_f^T)}{(N_s + N_f)} - \frac{tr((X_n X_n^T + X_f X_f^T) X^s X^{sT})}{N_s} > 0 \\
&\Leftrightarrow tr((N_s X_f X_f^T - N_f X^s X^{sT}) X_n X_n^T) > 0.
\end{aligned}$$

□

4.3 Experiments

In this section, we evaluate the proposed method for 2D object classification and face recognition. For object classification, we use the standard benchmark dataset *Office+Caltech* [77, 78] for domain adaptation. For face recognition, we follow [76] and conduct experiments on the CMU-PIE dataset [79]. We compare our method with several state-of-the-art unsupervised DA methods. Experimental results show that our method outperforms all other approaches significantly in most cases.

4.3.1 Object Recognition

Office+Caltech contains object images of four domains: Amazon (A), Webcam (W), DSLR (D), and Caltech (C). This leads to a total of 12 domain pairs for test. 10 common classes are selected in all domains. For each class, A, C, D and W have about 100, 100, 15 and 30 images, respectively. We follow the protocol used in [69, 80] to generate the source and target domain data. DeCAF features [81] are used in our experiment.

We compare two non-adaptation (NA) methods, and five state-of-the-art unsupervised DA methods: SVM and Dictionary Learning Based Classification (DLC) are the two NA methods, Subspace Interpolation via Dictionary Learning (SIDL) [76], Geodesic Flow Kernel (GFK) [66], Transfer Joint Matching (TJM) [80], Landmarks [69] and DA-NBNN [71] are the unsupervised DA methods. Dictionary trained using the DLC method is also used as the initial dictionary in our method.

GFK, SIDL and TJM are based on learning domain-invariant subspaces and they are fully unsupervised. In particular, SIDL shares a similar idea with GFS [65], but they use dictionary as basis. Landmarks reweight and select some source samples to assist adaptation, and they also utilize source labels to learn a discriminative classifier. DA-NBNN is a bootstrapping based method and is most closely related to our proposed approach, while our method differ from DA-NBNN in that we use different sample selection and stopping criteria.

We set $\lambda = 0.05$ and $\sigma^2 = 0.05$. For λ, [82] has shown it is non-sensitive to classification . For σ^2, we use maximum likelihood estimation to estimate it in

a similar way as suggested in [83] for each domain. In practice, we calculate the mean for all domains and set a uniform value for simplicity. For A, C, W and D, we set $K = 80, 80, 20$ and 8 respectively. Theoretically, Q can be set uniformly to 1. We can accelerate the convergence speed by setting Q to a reasonably larger value according to the dataset size. For A,C, W and D we set $Q = 8, 8, 2$ and 1, respectively. We only show the sensitivity analysis results on K and Q in section 4.3.

	Method	A→C	A→D	A→W	C→A	C→D	C→W	W→A	W→D	W→C	D→A	D→C	D→W
NA	SVM	85.04	87.90	78.98	91.44	89.81	80.00	75.68	99.36	71.95	87.06	78.81	98.64
	DLC	85.31	82.17	75.59	91.34	87.90	78.64	78.40	98.72	76.05	88.10	81.56	**99.32**
DA	GFK [66]	77.29	84.71	81.02	88.52	85.99	80.34	81.84	**100**	73.91	85.80	75.96	97.29
	SIDL [76]	84.51	81.53	74.24	90.92	89.81	78.31	75.05	**100**	71.15	87.89	80.14	**99.32**
	TJM [80]	80.14	84.71	75.25	89.04	85.35	76.94	84.86	**100**	78.01	87.37	77.38	98.64
	DA-NBNN [71]	83.44	80.89	76.61	89.67	87.90	80.34	88.00	**100**	82.46	91.34	86.11	97.97
	Landmarks [69]	84.68	85.99	82.37	92.38	**92.35**	84.07	84.03	98.73	71.68	77.04	74.35	95.25
	Proposed method	**86.73**	**92.36**	**88.47**	**93.31**	88.54	**95.59**	**92.80**	**100**	**88.69**	**93.11**	**89.13**	**99.32**

Table 4.1: Recognition accuracies on 12 pairs of cross-domain unsupervised object recognition. A: Amazon, C: Caltech, W: Webcam, D: DSLR

4.3.1.1 Results on recognition rate:

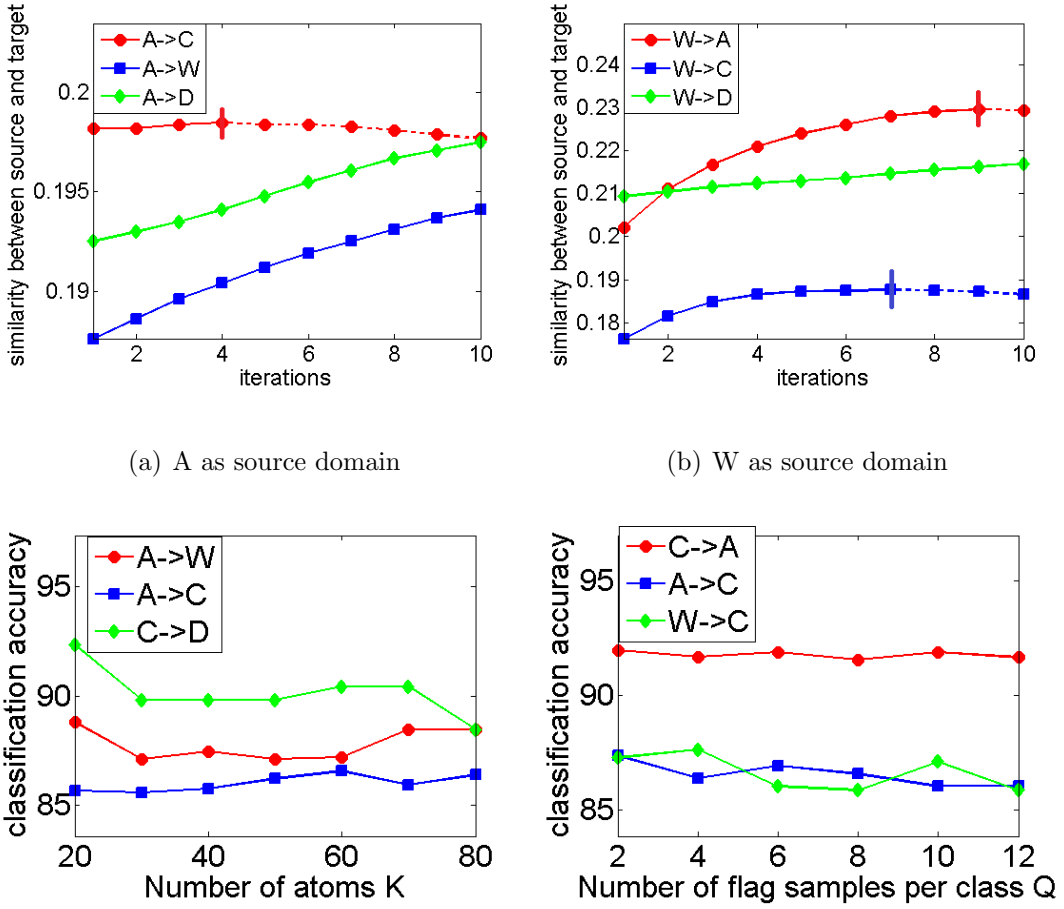
The recognition rates for all 12 domain pairs are summarized in Table 4.1. Our proposed approach outperforms other methods on most pairs by a large margin. We notice that the difficulty for the 12 adaptation tasks vary a lot. Our method tends to gain more over other approaches on more difficult pairs, *e.g.*, A→W, W→C, and

behaves similar to other methods on the easier pairs, *e.g.*, D→W. This indicates that our method can boost more on those pairs where domain dissimilarity is relatively large. The reason is that large domain discrepancy provides more scope for our adaptation process, which means adding the supportive samples can continuously reduce the domain divergence. In contrast, if the initial domain dissimilarity is small, adding the supportive samples may not reduce the domain distance in a significant way, and our method is likely to stop early and thus behave similar to other techniques.

We notice that [69] performs better than baselines when A or C act as the source domain. It demonstrates the effectiveness of selecting easier adaptive samples. However, its performance drops significantly when W or D act as the source domain. This is because when the source domain is relatively small, the selection of landmarks will further reduce the source domain size and leads to insufficient training data. In addition, the performance of [71] is good when W or D acts as the source domain. Thus it is very important to exploit the target discriminate information when the source domain is small.

4.3.1.2 Domain Similarity Evaluation:

In section 4.2.2, by setting up the stopping criterion, we proved that adding supportive samples reduce the domain divergence under a mild assumption. In this section, we compute the similarity of the source and target domains as the supportive samples are gradually added to the source domain. Results are shown in Fig. 4.2 (a) and (b). Here we set the adaptation iteration number to be 10 to monitor how



(a) A as source domain

(b) W as source domain

(c) dictionary atoms K

(d) Number of supportive samples per class Q

Figure 4.2: Domain similarity and parameter sensitivity. (a) and (b) show the change in domain similarity when the supportive samples are added to the source domain. Solid and dotted lines represent the iterations in which the domain similarity increases and decreases respectively. In our experiments, we only continue our adaptation as long as the similarity value goes up, which is represented by the solid lines before the slash symbols. (c) and (d) show the classification accuracy when K or Q varies. A: Amazon, C: Caltech, W: Webcam, D: DSLR

the similarity value changes as adaptation is performed. In our experiments, we only continue our adaptation as long as the similarity value goes up, which are represented by the solid lines. The dotted lines show that adding more supportive samples may enlarge the domain mismatch after some iterations. In this situation, the adaptation process should be terminated.

We compare the changes in domain similarity in Fig. 4.2 with our classification results in Table 4.1, and find that we are likely to gain more from our method when the domain similarity value continues to go up as more supportive samples are added to the source, *e.g.*, A→W. It indicates that reducing domain dissimilarity indeed helps the classification task.

It can be observed from Fig. 4.2 that when the domain similarity, before adaptation, is high it often means the NA methods can work well with high classification performance. However, in this case, as we add more supportive samples to the source domain, the domain similarity may change very little or even decrease, where the adaptation may bring no additional benefits or even harm the classification performance. In contrast, if the original domain similarity value is low, the condition in theorem 1 is easy to satisfy and the domain similarity can increase continuously as more supportive samples are added. Therefore, better results can be achieved as our adaptation process goes on. This explains why we can perform well in hard cases.

4.3.1.3 Parameter Sensitivity:

We conduct sensitivity analysis on parameter K and Q and show results on three pairs. Other pairs behave in a similar way. We can see from Figures. 4.2 (c) and (d) that the performance does not depend much on K and Q. Basically, a relatively small K makes the dictionary more compact and relatively a large Q accelerates the rate of convergence.

4.3.2 Face Recognition

Here we show the experimental results for face recognition on the CMU-PIE dataset. We follow the protocol presented in [76] and consider the proposed approach for face recognition under blur and illumination variations.

4.3.2.1 Across blur and illumination variance:

We select faces from 34 classes with 21 lighting conditions for each class, in which 11 samples for training and 10 samples for test. We add Gaussian blur and motion blur to test samples to evaluate different situations. Six situations are considered in our experiments: Gaussian blur with standard deviation of 3, 4, or 5, motion blur with lengths of 9, 11, or 13. In our experiments, λ is set to be 0.05. σ^2 is chosen to be 10. We set $K = 10$ and $Q = 1$. We compare our results with the same baseline methods as in section 4.3.1.

Results are presented in Table 4.2 and the proposed method outperforms other approaches by a large margin. We see that DLC approach gives us a good initial

Methods	$\sigma = 3$	$\sigma = 4$	$\sigma = 5$	$len = 9$	$len = 11$	$len = 13$
SVM	76.18	71.47	69.71	80.00	74.71	67.06
DLC	88.82	87.35	86.18	91.18	82.06	75.00
GFK [66]	78.53	77.65	74.71	84.41	73.82	64.71
SIDL [76]	80.29	77.94	76.76	85.88	81.18	73.53
TJM [80]	76.18	72.06	70.29	78.24	65.88	53.24
DA-NBNN [71]	62.35	58.53	57.94	65.59	54.12	42.65
Landmarks [69]	80.29	77.94	77.06	82.65	76.18	70.59
Proposed method	**94.70**	**93.24**	**90.29**	**96.47**	**93.24**	**92.35**

Table 4.2: Recognition accuracies on face recognition under illumination and blur mismatch.

point for adaptation. It indicates that dictionary-based classification methods are robust to Gaussian blur and motion blur, as well as illumination changes. We normally gain 5% -10% from the initial point and similar to object recognition, we tend to gain more when the initial mismatch between source and target is relatively large. Our method can overcome some blur variations at the beginning and then further reduce domain mismatch through adaptation from the source to target.

We can also interpret the physical meaning of the supportive sample faces. Since the light condition changes smoothly from source to target, the supportive samples should have closer illumination conditions with the source domain than other non-supportive samples. Once the supportive samples are added to the source domain, the rest of the samples in the target are easier to classify because the supportive samples reduce the illumination mismatch from source to target.

4.4 Conclusion

In this chapter, we proposed a novel incremental dictionary learning method for unsupervised domain adaptation. Supportive samples are iteratively selected to smoothly connect the source and target domains. We utilize the supportive samples to reduce the domain mismatch, and to build a more discriminate classifier, both of which are crucial for classification performance. We design an efficient stopping criterion to guarantee that adaptation reduces the domain dissimilarity monotonically. Extensive experiments on both object classification and face recognition datasets show promising results compared to many state-of-the-art DA methods.

Chapter 5: Unsupervised Domain-Specific Deblurring via Disentangled Representations

5.1 Overview

Image blur is an important factor that adversely affects the quality of images and thus significantly degrades the performances of many computer vision applications, such as object detection [32] and face recognition [23, 84]. To address this problem, blind image deblurring aims to restore the latent sharp image from a blurred image. Most conventional methods formulate the image deblurring task as a blur kernel estimation problem. Since this problem is highly ill-posed, many priors have been proposed to model the images and kernels [24–26]. However, most of these priors only perform well on generic natural images, but do not generalize to specific image domains, like face [27], text [28] and low-illumination images [29].

Recently, some learning-based approaches have been proposed for blind image restoration [27,32,33]. CNN-based models can handle more complex blur types and have enough capacity to train on large-scale datasets. Meanwhile, the Generative Adversarial Networks (GAN) have been found to be effective in generating more realistic images. Nonetheless, most of these methods need paired training data, which

(a) Blurred (b) [85] (c) Ours (d) Blurred (e) [86] (f) Ours

Figure 5.1: Qualitative deblurred results of the proposed method compared with other state-of-the-art unpaired deblurring methods on real-world blurred face and text images.

is expensive to collect in practice. Although numerous blur generation methods have been developed [32, 34, 35], they are not capable of learning all types of blur variants in the wild. Moreover, strong supervision may cause algorithms to overfit training data and thus cannot generalize well to real images.

More recently, Nimisha *et al.* [85] proposed an unsupervised image deblurring method based on GANs where they add reblur loss and multi-scale gradient loss on the model. Although they achieved good performance on synthetic datasets, their results on some real blurred images are not satisfactory (Fig. 5.1(b)). Another solution might be to directly use some existing unsupervised methods (CycleGAN [86], DualGAN [87]) to learn the mappings between sharp and blurred images. However, these generic approaches often encode other factors (*e.g.*, color, texture) rather than blur information into the generators, and thus do not produce good restored images (Fig. 5.1(e)).

In this chapter, we propose an unsupervised domain-specific image deblurring method based on disentangled representations. More specifically, we disentangle the

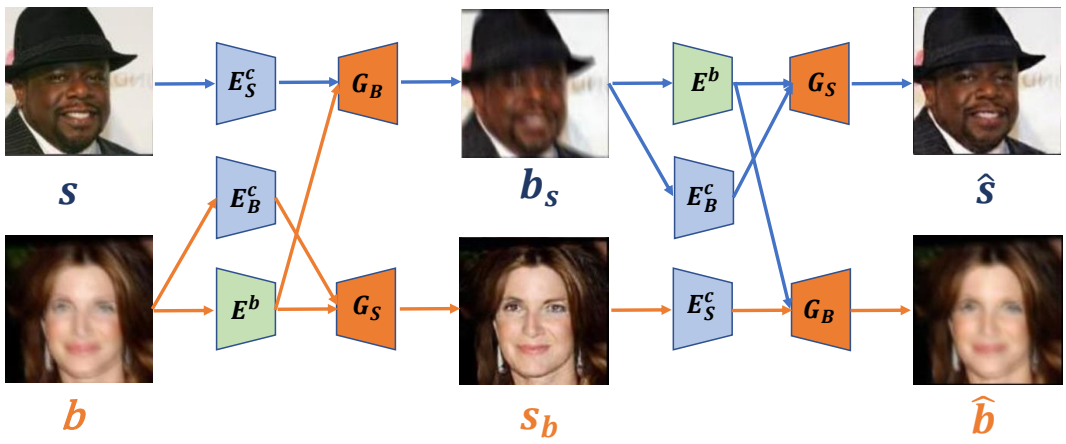


Figure 5.2: Overview of the deblurring framework.The data flow of the top *blurring* branch (bottom *deblurring* branch) is represented by blue (orange) arrows. E_B^c and E_S^c are content encoders for blurred and sharp images respectively; E^b is blur encoder; G_B and G_S represent blurred image and sharp image generators respectively. Two GAN losses are added to distinguish b_s from blur images, and to distinguish s_b from sharp images. The KL divergence loss is added to the output of E^b. Cycle-consistency loss is added to s and $\hat{s}$, b and $\hat{b}$. Perceptual loss is added to b and s_b.

content and blur features from blurred images to accurately encode blur information into the deblurring framework. As shown in Fig. 5.2, the content encoders extract content features from unpaired sharp and blurred images, and the blur encoder captures blur information. In addition, we share the weights of the last layer of both content encoders so that the content encoders can project the content features of both domains onto a common space. However, this structure by itself does not guarantee that the blur encoder captures blur features—it may encode content features as well. Inspired by [88], we add a KL divergence loss to regularize the distribution of blur features to suppress the contained content information. Then, the deblurring generator G_S and the blurring generator G_B take corresponding content features conditioned on blur attributes to generate deblurred and blurred images. Similar to CycleGAN [86], we also use the adversarial loss and the cycle-consistency loss as regularizers to assist the generator networks to yield more realistic images, and also preserve the content of the original image. To further remove the unpleasant artifacts introduced by the deblurring generator G_S, we add the perceptual loss to the proposed framework. Some sample deblurred images are shown in Fig. 5.1.

We conduct extensive experiments on face and text deblurring and achieve competitive performance compared with other state-of-the-art deblurring methods. We also evaluate the proposed method on face verification and optical character recognition (OCR) tasks to demonstrate the effectiveness of our algorithm on recovering semantic information.

5.2 Proposed Method

The proposed approach consists of four parts: 1) content encoders E_B^c and E_S^c for blurred and sharp image domains; 2) blur encoder E^b; 3) blurred and sharp image generators G_B and G_S; 4) blurred and sharp image discriminators D_B and D_S. Given a training sample $b \in B$ in the blurred image domain and $s \in S$ in the sharp image domain, the content encoders E_B^c and E_S^c extract content information from corresponding samples and E^b estimates the blur information from b. G_S then takes $E_B^c(b)$ and $E^b(b)$ to generate a sharp image s_b while G_B takes $E_S^c(s)$ and $E^b(b)$ to generate a blurred image b_s. The discriminators D_B and D_S distinguish between the real and generated examples. The end-to-end architecture is illustrated in Fig. 5.2.

In the following subsections, we first introduce the method to disentangle content and blur components in Section 5.2.1. Then, we discuss the loss functions used in our approach. In Section 5.2.5, we describe the testing procedure of the proposed framework. Finally, the implementation details are discussed in Section 5.2.6.

5.2.1 Disentanglement of Content and Blur

Since the ground truth sharp images are not available in the unpaired setting, it is not trivial to disentangle the content information from a blurred image. However, since sharp images only contain content components without any blur information, the content encoder E_S^c should be a good content extractor. We enforce the last layer of E_B^c and E_S^c to share weights so as to guide E_B^c to learn how to effectively

extract content information from blurred images.

On the other hand, the blur encoder E^b should only encode blur information. To achieve this goal, we propose two methods to help E^b suppress as much content information as possible. First, we feed $E^b(b)$ together with $E^c_S(s)$ into G_B to generate b_s. Since b_s is a blurred version of s and it will not contain content information of b, this structure discourages $E^b(b)$ to encode content information of b. Second, we add a KL divergence loss to regularize the distribution of the blur features $z_b = E^b(b)$ to be close to the normal distribution $p(z) \sim N(0, 1)$. As shown in [88], this will further suppress the content information contained in z_b. The KL divergence loss is defined as follows:

$$KL(q(z_b)||p(z)) = -\int q(z_b) \log \frac{p(z)}{q(z_b)} dz \tag{5.1}$$

As proved in [26], minimizing the KL divergence is equivalent to minimizing the following loss:

$$\mathcal{L}_{KL} = \frac{1}{2} \sum_{i=1}^{N} (\mu_i^2 + \sigma_i^2 - \log(\sigma_i^2) - 1) \tag{5.2}$$

where μ and σ are the mean and standard deviation of z_b and N is the dimension of z_b. Similar to [26], z_b is sampled as $z_b = \mu + z \circ \sigma$, where $p(z) \sim N(0, 1)$ and $\circ$ represents element-wise multiplication.

5.2.2 Adversarial Loss

In order to make the generated images look more realistic, we apply the adversarial loss on both domains. For the sharp image domain, we define the adversarial

loss as:

$$\begin{aligned}\mathcal{L}_{D_S} = \quad & \mathbb{E}_{s\sim p(s)}[\log D_S(s)] \\ & + \mathbb{E}_{b\sim p(b)}[\log(1 - D_S(G_S(E_B^c(b), z_b)))]\end{aligned} \tag{5.3}$$

where D_S tries to maximize the objective function to distinguish between restored and real sharp images. In contrast, G_S aims to minimize the loss to make deblurred images look similar to real samples in domain S. Similarly, we define the adversarial loss in blurred image domain as $\mathcal{L}_{D_B}$:

$$\begin{aligned}\mathcal{L}_{D_B} = \quad & \mathbb{E}_{b\sim p(b)}[\log D_B(b)] \\ & + \mathbb{E}_{s\sim p(s)}[\log(1 - D_B(G_B(E_S^c(s), z_b)))]\end{aligned} \tag{5.4}$$

5.2.3 Cycle-Consistency Loss

After competing with discriminator D_S in the minmax game, G_S should be able to generate visually realistic sharp images. However, since no pairwise supervision is provided, the deblurred image may not retain the content information in the original blurred image. Inspired by CycleGAN [86], we introduce the cycle-consistency loss to guarantee that the deblurred image s_b can be reblurred to reconstruct the original blurred sample, and b_s can be translated back to the original sharp image domain. The cycle-consistency loss further limits the space of the generated samples and preserves the content of original images. More specifically, we perform the forward translation as:

$$s_b = G_S(E_B^c(b), E^b(b)), b_s = G_B(E_S^c(s), E^b(b)) \tag{5.5}$$

and the backward translation as:

$$\hat{b} = G_B(E_S^c(s_b), E^b(b_s)), \hat{s} = G_S(E_B^c(b_s), E^b(b_s)) \tag{5.6}$$

We define the cycle-consistency loss on both domains as:

$$\mathcal{L}_{cc} = \mathbb{E}_{s\sim p(s)}[\|s - \hat{s}\|_1] + \mathbb{E}_{b\sim p(b)}[||b - \hat{b}||_1] \tag{5.7}$$

5.2.4 Perceptual Loss

From our preliminary experiments, we find that the generated deblurred samples often contain many unpleasant artifacts. Motivated by observations from [89,90] that features extracted from pre-trained deep networks contain rich semantic information, and their distances can act as perceptual similarity judgments, we add a perceptual loss between the deblurred images and corresponding original blurred images:

$$\mathcal{L}_p = \|\phi_l(s_b) - \phi_l(b)\|_2^2 \tag{5.8}$$

where $\phi_l(x)$ is the features of the l-th layer of the pre-trained CNN. In our experiments, we use the $\texttt{conv}_{3,3}$ layer of VGG-19 network [91] pre-trained on ImageNet [92].

There are two main reasons why we use the blurred image b instead of the sharp one s as the reference image in the perceptual loss. First, we assume that the content information of b can be extracted by the pre-trained CNN. As shown in Fig. 5.3.2, the experimental results confirm this point. Second, since s and b are unpaired, applying the perceptual loss between s and s_b will force s_b to encode irrelevant content information from s. However, since we also notice that the perceptual loss is sensitive to blur as shown in [93], we carefully balance the weight of the perceptual

loss with other losses to prevent s_b from staying too close to b. The sensitivity evaluation of varying the weight is conducted in Section 5.3.3.

It is worth mentioning that the perceptual loss is not added to b_s and s. This is because we do not find obvious artifacts in b_s during training. Moreover, for text image deblurring, since we observe the percetual loss does not help but sometimes hurt the performance, we do not include it for this task. One possible reason may be due to the pixel intensity distribution of the text images being very different from the natural images in the ImageNet dataset.

The full objective function is a weighted sum of all the losses from (5.2) to (5.8):

$$\mathcal{L} = \lambda_{adv}\mathcal{L}_{adv} + \lambda_{KL}\mathcal{L}_{KL} + \lambda_{cc}\mathcal{L}_{cc} + \lambda_{p}\mathcal{L}_{p} \tag{5.9}$$

where $\mathcal{L}_{adv} = \mathcal{L}_{D_S} + \mathcal{L}_{D_B}$. We empirically set the weights of each loss to balance their importances.

5.2.5 Testing

At test time, the blurring branch is removed. Given a test blurred image b_t, E_B^c and E^b extract the content and blur features. Then G_S takes the outputs and generates the deblurred image s_{b_t}:

$$s_{b_t} = G_S(E_B^c(b_t), E^b(b_t)) \tag{5.10}$$

5.2.6 Implementation Details

Architecture and training details. For the network architectures, we follow the structures similar to the one used in [94]. The content encoder is composed of three strided convolution layers and four residual blocks. The blur encoder contains four strided convolution layers and a fully connected layer. For the generator, the architecture is symmetric to the content encoder with four residual blocks followed by three transposed convolution layers. The discriminator applies a multi-scale structure where feature maps at each scale go through five convolution layers and then are fed into sigmoid outputs. The end-to-end design is implemented in PyTorch [95]. During training, we use Adam solver [96] to perform two steps of update on discriminators, and then one step on encoders and generators. The learning rate is initially set to 0.0002 for the first 40 epochs, then we use exponential decay over the next 40 epochs. In all the experiments, we randomly crop 128×128 patches with batch size of 16 for training. For hyper-parameters, we experimentally set: $\lambda_{adv} = 1$, $\lambda_{KL} = 0.01$, $\lambda_{cc} = 10$ and $\lambda_p = 0.1$.

Motion blur generation. We follow the procedure in DeblurGAN [32] to generate motion blur kernels to blur face images. A random trajectory is generated as described in [97]. Then the kernels are generated by applying sub-pixel interpolation to the trajectory vector. For parameters, we use the same values as in [32] except that we set the probability of impulsive shake as 0.005, the probability of Gaussian shake uniformly distributed in $(0.5, 1.0)$, and the max length of the movement as 10.

5.3 Experimental Results

We evaluate the proposed approach on three datasets: CelebA dataset [98], BMVC_Text dataset [28], and CFP dataset [99].

5.3.1 Datasets and Metrics

CelebA dataset: This dataset consists of more than 202,000 face images. Most of the faces are of good quality and at near-frontal poses. We randomly split the whole dataset into three mutually exclusive subsets: sharp training set (100K images), blurred training set (100K images) and test set (2137 images). For the blurred training set, we use the method in Section 5.2.6 to blur the images. The faces are detected and aligned using the method proposed in [44].

BMVC_text dataset: This dataset is composed of 66,000 text images with size 300×300 for training and 94 images with size 512×512 for OCR testing. Similar to CelebA, we evenly split the training sets as sharp and blurred set. Since the dataset already contains the blurred text images, we directly use them instead of generating new ones.

CFP dataset: This dataset consists of 7,000 still images from 500 subjects and for each subject, it has ten images in frontal pose and four images in profile pose. The datasets are divided into ten splits and two protocols: frontal-to-frontal (FF) and frontal-to-profile (FP). We used the same method as described above to blur the images. The faces are detected and aligned similarly as the CelebA dataset.

For CelebA and BMVC_Text datasets, we use standard debluring metrics

(PSNR, SSIM) for evaluation. We also use feature distance (*i.e.*, the L_2 distance of the outputs from some deep networks) between the deblurred image and the ground truth image as a measure of semantic similarity because we find this to be a better perceptual metric than PSNR and SSIM [93]. For the CelebA dataset, we use the outputs of `pool5` layer from VGG-Face [100] and for the text dataset, we use the outputs of `pool5` layer from a VGG-19 network. For text deblurring, another meaningful metric is the OCR recognition rate for the deblurred text. We follow the same protocol as in [28] to report the character error rate (CER) for OCR evaluation.

To study the influence of motion blur on face recognition and test the performance of different deblurring algorithms, we perform face verification on the CFP dataset. Both frontal-to-frontal and frontal-to-profile protocol are evaluated. The frontal-to-profile protocol can further be used to examine the robustness of the deblurring methods on pose.

In order to test the generalization capability of the proposed method, we also try our approach on natural images. More details are presented in the supplementary materials.

5.3.2 Ablation Study

In this section, we present the results of an ablation study to analyze the effectiveness of each component or loss in the proposed framework. Both quantitative and qualitative results on CelebA dataset are reported for the following five variants of our methods where each component is gradually added: 1) only including

Figure 5.3: Ablation study. (a) shows the blurred image and (g) is the sharp image. (b) only contains deblurring branch (bottom branch of Fig. 5.2), (c) adds blurring branch (bottom branch of Fig. 5.2), (d) adds disentanglement (E^b), (e) adds the KL divergence loss, and (f) adds perceptual loss.

deblurring branch (*i.e.*, removing the top cycle in Fig. 5.2 and the blur encoder E^b); 2) adding blurring branch (adding the top cycle of Fig. 5.2); 3) adding content and blur disentanglement; 4) adding the KL divergence loss; 5) adding the perceptual loss.

We present the PSNR, SSIM and VGG-Face distance (d_{VGG}) for each variant in Table 5.1 and the visual comparisons are shown in Fig. 5.3. From Table 5.1, we see that adding the blurring branch significantly improves the deblurring performance, especially for the perceptual distance. As shown in Fig. 5.3 (c) many artifacts are removed from face and colors are preserved well compared to (b). This confirms the findings in CycleGAN [86] that only one direction cycle-consistency loss is not enough to recover good images. However, we find that adding a disentanglement component does not help but rather hurt the performance (Fig. 5.3 (d)). This demonstrates that the blurring encoder E^b will induce some noise and confuse the generator G_S if the KL divergence loss is not enforced. In contrast, when the

Method	PSNR	SSIM	d_{VGG}
Only deblurring branch	18.83	0.56	82.9
Add blurring branch	19.84	0.59	65.5
Add disentanglement	19.58	0.57	69.8
Add KL divergence loss	20.29	0.61	60.6
Add perceptual loss	**20.81**	**0.65**	**57.6**

Table 5.1: Ablation study on the effectiveness of different components. d_{VGG} represents the distance of feature from VGG-Face, lower is better.

KL diveregence loss is added to E^b (Fig. 5.3 (e)), content and blur information can be better disentangled and we observe some improvements on both PSNR and perceptual similarities. Finally, the perceptual loss can improve the perceptual reality of the face notably. By comparing Fig. 5.3 (e) and (f), we find that the artifacts on cheek and forehead are further removed. Furthermore, the mouth region of (f) is more realistic than (e).

5.3.3 Paramter selection for λ_p

As we mentioned above, the weight for perceptual loss λ_p needs to be tuned so that the deblurred image neither stays too close to the original blurred image, nor contains many artifacts. The quantitative performance and qualitative visualizations are shown in Table 5.2 and Fig. 5.4 respectively. If setting the λ_p too high ($\lambda_p = 1$), the deblurred images become very blurred (Fig. 5.4(b)), and both the

Values	PSNR	SSIM	d_{VGG}
$\lambda_p = 1$	18.40	0.59	78.0
$\lambda_p = 0.1$	**20.81**	**0.65**	**57.6**
$\lambda_p = 0.01$	20.21	0.62	58.7

Table 5.2: Quantitative results for different settings of λ_p.

(a) Blurred (b) $\lambda_p = 1$ (c) $\lambda_p = 0.1$ (d) $\lambda_p = 0.01$ (e) Sharp

Figure 5.4: Visualizations of sample images with different settings of λ_p. Best viewed by zooming in.

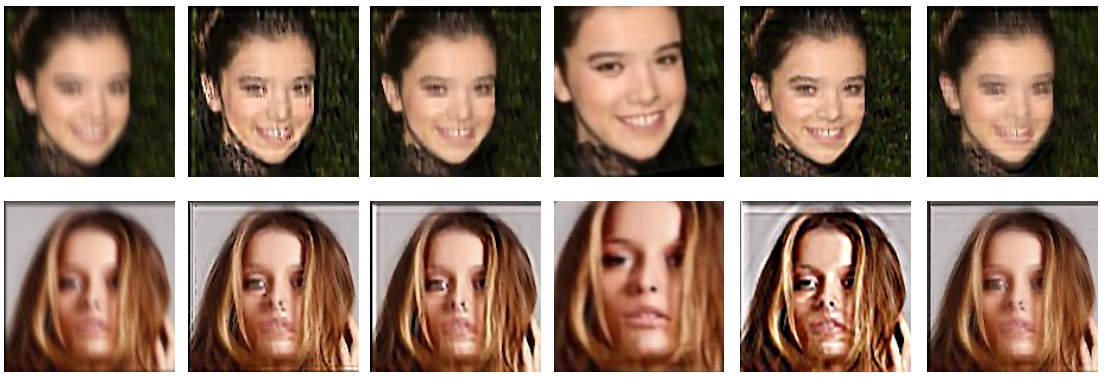
quantitative performance and visualization results are poor. In contrast, if λ_p is set too low ($\lambda_p = 0.01$), the deblurred images contain many artifacts (Fig. 5.4(d)).

5.3.4 Face Results

Compared methods: We compare the proposed method with some state-of-the-art deblurring methods [24, 25, 27, 30–33, 86, 101]. We directly use the pre-trained models provided by authors except for CycleGAN [86], where we retrain the model by using the same training set as our method. Both CNN-based models

Method	PSNR	SSIM	d_{VGG}
Pan *et al.* [30]	17.34	0.52	96.6
Pan *et al.* [24]	17.59	0.54	85.6
Shen *et al.* [27]	**21.50**	**0.69**	**57.9**
Pan *et al.* [31]	15.16	0.38	166.6
Xu *et al.* [25]	16.84	0.47	102.0
Krishnan *et al.* [101]	18.51	0.56	89.4
Kupyn *et al.* [32]	18.86	0.54	116.5
Nah *et al.* [33]	18.26	0.57	75.6
Zhu *et al.* [86]	19.40	0.56	103.2
Ours	**20.81**	**0.65**	**57.6**

Table 5.3: Quantitative performance comparison with state-of-the-art methods on CelebA dataset. d_{VGG} represents the distance of feature from VGG-Face, lower is better.



(a) Blurred (b) [30] (c) [24] (d) [27] (e) [31] (f) [25]

(g) [101] (h) [32] (i) [33] (j) [86] (k) Ours (l) Sharp

Figure 5.5: Visual performance comparison with state-of-the-art methods on CelebA dataset. Best viewed in color and by zooming in.

[27,32,33,86] and conventional MAP-based methods are included [24,25,30,31,101]. Among these approaches, two are specific for face deblurring [27, 31] while others are generic deblurring algorithm. The kernel size for [24, 30] is set to 9. We found that the face deblurring method [27] is very sensitive to face alignment, we follow the sample image provided by the author to align the faces before running their algorithm. Meanwhile, CycleGAN is the only unsupervised CNN-based method we compare with.

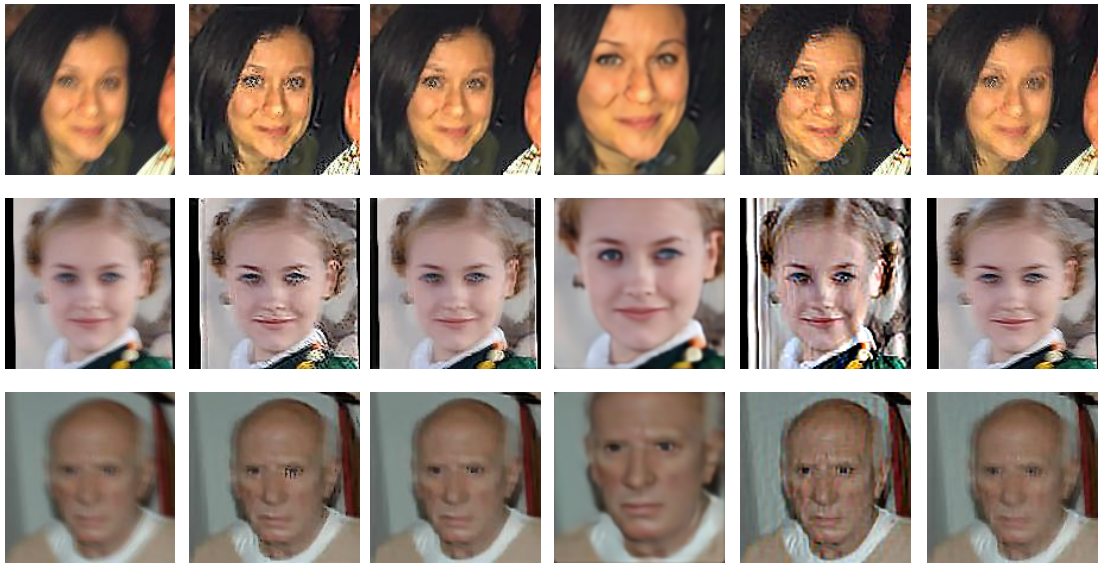
CelebA dataset results. The quantitative results for CelebA dataset are shown in Table 5.3 and the visual comparisons are illustrated in Fig. 5.5. Our

Methods	F2F Accuracy	F2P Accuracy
Blurred	0.920±0.014	0.848±0.013
Sharp	0.988±0.005	0.949±0.014
Pan *et al.* [30]	0.930±0.013	0.853±0.010
Pan *et al.* [24]	0.935±0.015	0.872±0.015
Shen *et al.* [27]	0.959±0.008	0.821±0.022
Pan *et al.* [31]	0.916±0.011	0.825±0.016
Xu *et al.* [25]	0.944±0.012	0.865±0.013
Krishnan *et al.* [101]	0.941±0.012	0.857±0.014
Kupyn *et al.* [32]	0.948±0.012	0.872±0.007
Nah *et al.* [33]	**0.960±0.007**	**0.885±0.016**
Zhu *et al.* [86]	0.941±0.012	0.864±0.015
Ours	**0.948±0.006**	**0.872±0.015**

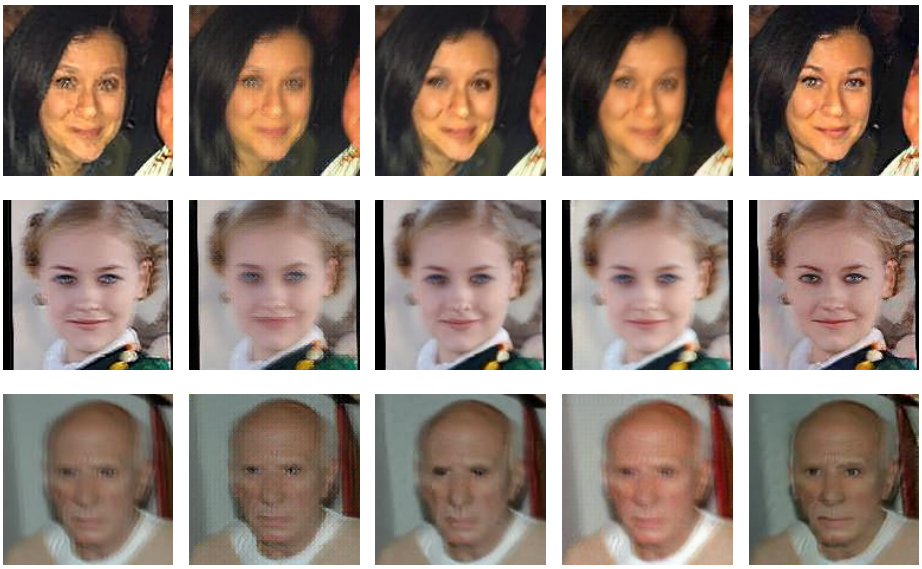
Table 5.4: Face verification results on the CFP dataset. F2F, F2P represent frontal-to-frontal and frontal-to-profile protocols.

approach shows superior performance to other unsupervised algorithms on both conventional metrics and VGG-Face distance. Furthermore, we achieve comparable results with state-of-the-art supervised face deblurring method [27]. From Fig. 5.5 we see that conventional methods often over-deblur or under-deblur the blurred images. Among them, Krishnan *et al.* [101] perform the best in PSNR and SSIM and Pan *et al.* [24] perform the best in perceptual distance. For CNN-based methods, Shen *et al.* [27] include a face parsing branch and achieve the best performance among the compared methods. The results for DeblurGAN [32] contain some ringing artifacts and CycleGAN [86] cannot recover the mouth part of both images that well. Nah *et al.* [33] shows better visual results than other CNN-based generic methods but still contains some blur in local structures.

Face verification results. The face verification results for the CFP dataset are reported in Table 5.4. We train a 27-layer ResNet [23] on the curated MS-Celeb1M dataset [15, 16] with 3.7 millions face images and extract features of the deblurred faces for each method. Cosine similarities of test pairs are used as similarity scores for face verification. We follow the protocols used in [9, 102] and the verification accuracy for both frontal-to-frontal and frontal-to-profile protocols are reported. As shown in Table 5.4, the proposed method improves the baseline results of blurred images and outperforms CycleGAN [86] on both protocols. Moreover, we achieve comparable performance compared to other state-of-the-art supervised deblurring methods. Shen *et al.* [27] perform very well for frontal-to-frontal protocol, yet provide the worst performance on frontal-to-profile protocol, which shows that the face parsing network in their method is sensitive to poses. In contrast, the



(a) Blurred (b) [30] (c) [24] (d) [27] (e) [31] (f) [25]



(g) [101] (h) [32] (i) [33] (j) [86] (k) Ours

Figure 5.6: Visual comparisons with state-of-the-art methods on real blurred face images. Best viewed in color and by zooming in.

proposed method works for both frontal and profile face images even though we do not explicitly train on faces with extreme poses.

Real blurred images results We also evaluate the proposed method on some real-world images from the datasets of Lai *et al.* [103], and the results are shown in Fig. 5.6. Similar to what we have observed for CelebA, our method shows competitive performance compared to other state-of-the-art approaches. Conventional methods [24, 25, 30, 31, 101] still tend to under-deblur or over-deblur images, especially on local regions such as eyes and mouths. On the other hand, the generic CNN-based method [32] does not perform very well on face deblurring. CycleGAN [86] fails to recover sharp faces but only changes the background color of images (*e.g.*, third row of Fig. 5.6(j)). Nah *et al.* [33] produce good results on the first two faces, but generate some artifacts in the third image. Deep semantic face deblurring [27] generates better results than other compared methods. Nonetheless, due to the existence of face parsing, they tend to sharpen some facial parts (eye, nose and mouth) but over-smooth the ears and the background. In contrast, our method not only recovers sharp faces, but also restores sharp textures in the background (*e.g.*, third row of Fig. 5.6(k)).

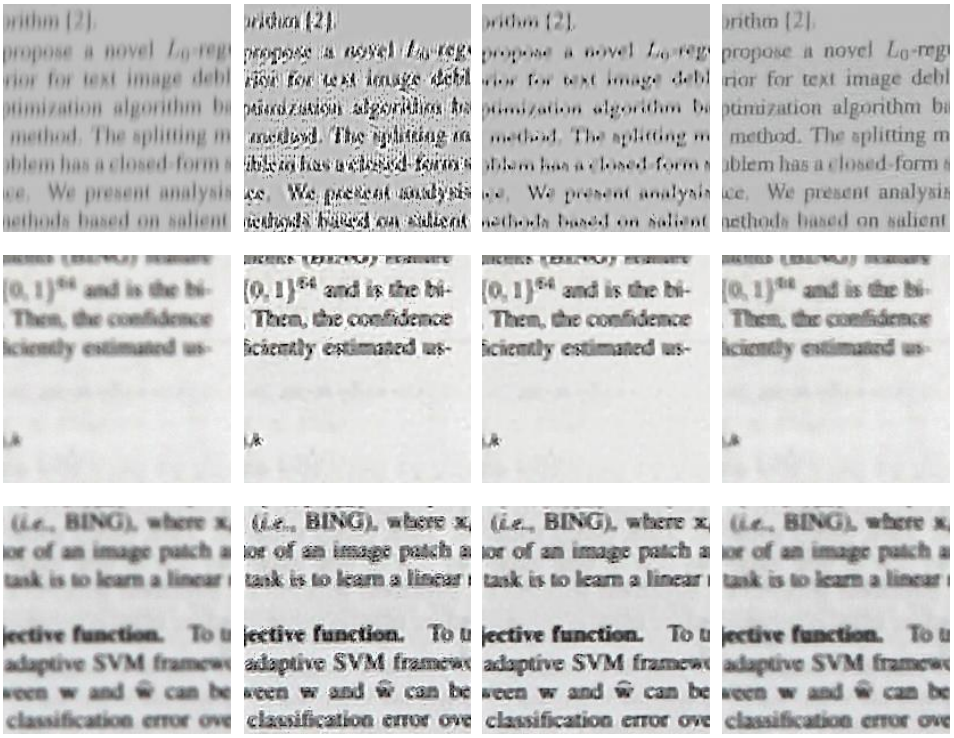
5.3.5 Text results

BMVC_Text dataset results. Similar to face experiments, we train a CycleGAN model using the same training set as our method. The kernel size for [24,30] is set to 12. The quantative results for the BMVC_Text dataset are shown in Table 5.5

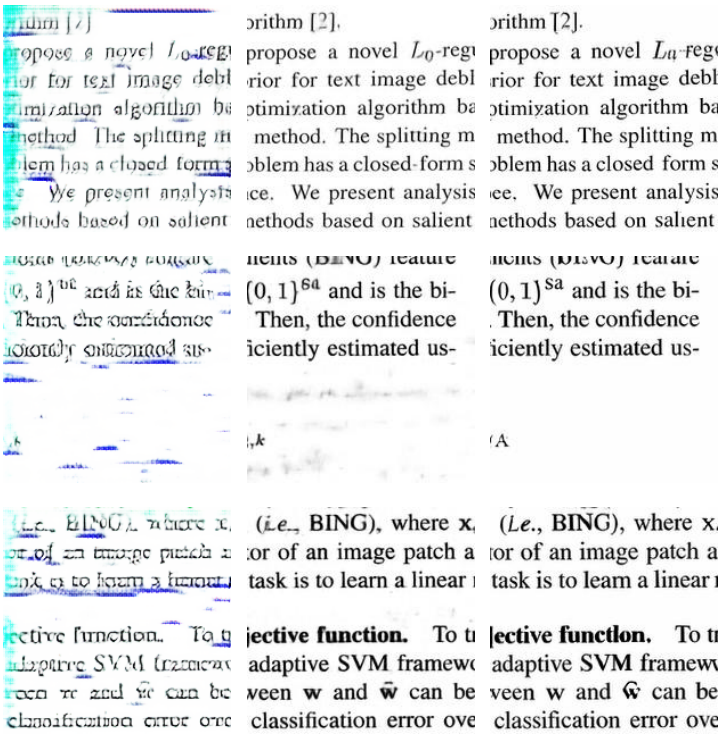
(a) Blurred (b) [30] (c) [24] (d) [33]

(e) [86] (f) [28] (g) Ours (h) Sharp

Figure 5.7: Visual results compared with state-of-the-art methods on BMVC_Text dataset. Best viewed by zooming in.



(a) Blurred (b) [30] (c) [24] (d) [33]



(e) [86] (f) [28] (g) Ours

Figure 5.8: Visual results compared with state-of-the-art methods on real blurred text images. Best viewed by zooming in.

Method	PSNR	SSIM	d_{VGG}	CER
Pan *et al.* [30]	21.18	0.92	19.7	42.3
Pan *et al.* [24]	21.84	0.93	15.7	35.3
Nah *et al.* [33]	22.27	0.92	31.9	50.6
Hradis *et al.* [28]	**30.6**	**0.98**	**1.6**	**7.2**
Zhu *et al.* [86]	19.57	0.89	18.8	53.0
Ours	**22.56**	**0.95**	**2.2**	**10.1**

Table 5.5: Quantitative performance comparison with state-of-the-art methods on BMVC_Text dataset. d_{VGG} represents the distance of feature from VGG-Face, lower is better. CER is the OCR character error rate, lower is better.

and some sample images are presented in Fig. 5.7. We can see that conventional methods [24, 30] and generic deblurring approaches [33] do not perform well on text deblurring. The visual quality is poor and the OCR error rate is very high. The results for CycleGAN [86] contain some unexplainable blue background. Although it removes the blur in images, it fails to recover recognizable text. In contrast, our method achieves good visual quality and its performance is comparable to the state-of-the-art supervised text deblurring method [28] on semantic metrics (*i.e.*, perceptual distance and OCR error rate). Interestingly, we find the PNSR performance for our approach is worse than the method [28] by large margins. We carefully examine our visual results and find that the proposed method sometimes changes the font of the text while deblurring. For example, as shown in the first row of Fig. 5.7(g),

the font of our deblurred text becomes lighter and thinner compared to the original sharp text image (Fig. 5.7(h)). The main reason for this phenomenon is that our method does not utilize paired training data so that the deblurring generator cannot preserve some local details of text images.

Real blurred text images results We also evaluate our deblurring method on real blurred text images provided by Hradis *et al.* [28]. Due to space limitation, 200×200 patches are randomly cropped, and some visual results are illustrated in Fig. 5.8. Similar to the results of BMVC_Text dataset, we find that conventional methods [24,30] fail to deblur the given text images. Nah *et al.* [33], in contrast, generate a reasonable deblurred result for the first image but cannot handle the second one. CycleGAN [86] again produces blue artifacts and cannot recover meaningful text information. Hradis*et al.* [28] and our approach both generate satisfactory results. Although we mis-recognize some characters (*e.g.*, in the second images, "*i.e.*, BING" is recovered as "*Le.*,BING"), we still correctly recover most of the blurred images.

5.4 Conclusions

In this chapter, we presented an unsupervised method for domain-specific single image deblurring. We disentangled the content and blur features in a blurred image and added the KL divergence loss to discourage the blur features to encode content information. In order to preserve the content of the original images, we added a blurring branch and cycle-consistency loss to the framework. The percep-

tual loss helps the blurred image remove unrealistic artifacts. Ablation study on each component shows the effectiveness of different modules. We conducted extensive experiments on face and text deblurring. Both quantative and visual results show promising performance compared to other state-of-the-art approaches.

Chapter 6: Regularized Metric Adaptation for Unconstrained Face Verification

6.1 Overview

Face verification research has been one of the active research areas in computer vision community for decades. Although the performance on the constrained face verification dataset has been already pushed to surpass human performance, the problem of unconstrained face verification under extreme pose, illumination, and expression variations is still unsolved. Moreover, the acquisition condition of the training samples may not match the condition of the test pairs, which may lead to the domain mismatch problem. In this chapter, we propose a metric adaptation method for the template-based face verification problem. Given a pair of templates, the idea of metric adaptation is to learn a template-specific metric by utilizing the intra-information between features in one template and the inter-information between the template and the negative set (*i.e.*, the negative set consists of samples from subjects who are mutually exclusive to the test data.). In principle, this is similar to the one-shot approach [36]. However, one-shot learning methods mainly consider one-to-one verification where intra-information inside the templates cannot

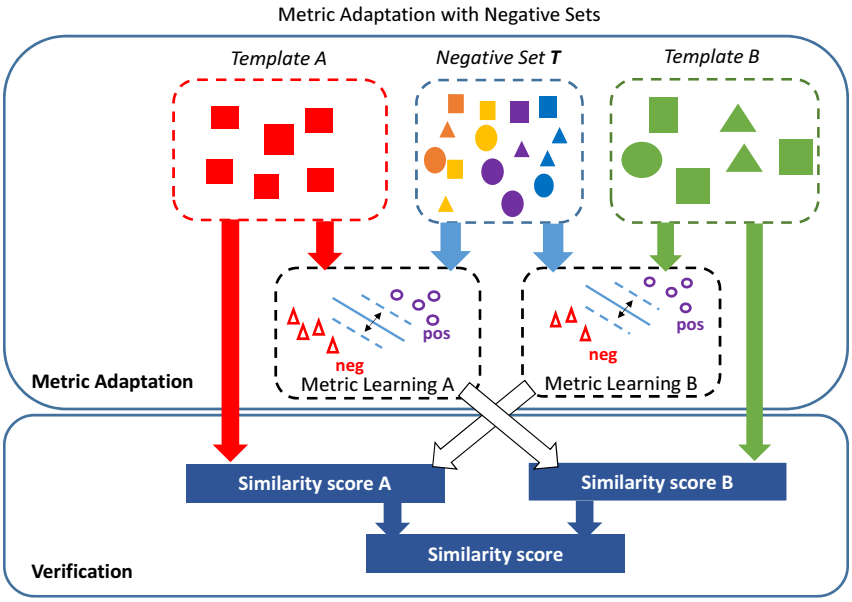


Figure 6.1: The system overview of the proposed regularized metric adaptation method for unconstrained face verification.

be exploited.

In general, the proposed regularized JBML framework not only alleviates the over-fitting problem, but also provides a way to significantly reduce the model size without much degradation in performance. We also analyze the selection of the negative set to reduce its size and to accelerate the metric learning process. Extensive experiments on IJB-A and CS2 datasets yield promising results compared to other competitive methods.

6.2 Proposed Method

6.2.1 Regularized Joint Bayesian Metric Learning

The joint Bayesian metric learning has been shown to be effective for face verification [58, 104]. Its formulation can also be interpreted as the combination of two components: Mahalanobis distance and projected cosine similarity. In general, directly minimizing the hinge loss objective function usually results in a large model complexity and over-fitting problems due to a large number of parameters introduced by metric matrices. On the other hand, Euclidean distance and cosine similarity provide a good baseline performance on deep convolutional features [104] for the face verification task. In addition, Euclidean distance and cosine similarity have better generalization capability because they are not trained on a particular training set. The model size for Euclidean and cosine metric is also small since only the diagonal terms are non-zeros. Therefore, we add the regularization terms to enforce the learned metric matrices to stay close to identity matrices, since when both metric matrices are identity, the computation of the similarity scores reduces to the summation of the Euclidean distance and the cosine similarity.

Given a set of features $\mathbf{X}$, we construct positive pairs if both features belong to the same person and negative pairs otherwise. The goal of the metric learning is to increase the similarity score of positive pairs while decreasing the negative ones.

We solve an optimization problem as follows:

$$\underset{\mathbf{W},\mathbf{V},b}{\operatorname{argmin}} \sum_{ij} \max\{0, \alpha - l_{ij}(b - d_{\mathbf{W}}(x_i, x_j) + 2s_{\mathbf{V}}(x_i, x_j))\} + \lambda_1 \|\mathbf{W} - \mathbf{I}\|_F^2 + \lambda_2 \|\mathbf{V} - \mathbf{I}\|_F^2 \tag{6.1}$$

where $d_{\mathbf{W}}(x_i, x_j) = (x_i - x_j)^T \mathbf{W}^T \mathbf{W}(x_i - x_j)$ is the Mahalanobis distance and $s_{\mathbf{V}}(x_i, x_j) = x_i^T \mathbf{V}^T \mathbf{V} x_j$ is the projected similarity. Both $\mathbf{W} \in \mathbb{R}^{d \times d}$ and $\mathbf{V} \in \mathbb{R}^{d \times d}$ are the projection matrices. $l_{ij} = 1$ if $\{x_i, x_j\}$ is a positive pair and $l_{ij} = -1$, otherwise. b is the bias and α is the margin parameter. λ_1, λ_2 are the regularization parameters to control the regularization terms.

To solve the optimization problem in (6.1), we apply the SGD method as follows:

$$\begin{aligned}
\mathbf{W}_{t+1} &= \begin{cases} \mathbf{W}_t, & \text{if } l_{ij}\rho_{ij} \geq \alpha \\ \mathbf{W}_t - \tau(l_{ij}\mathbf{W}_t\mathbf{\Psi}_{ij} + \lambda_1(\mathbf{W} - \mathbf{I})), & \text{otherwise,} \end{cases} \\
\mathbf{V}_{t+1} &= \begin{cases} \mathbf{V}_t, & \text{if } l_{ij}\rho_{ij} \geq \alpha \\ \mathbf{V}_t + \tau(l_{ij}\mathbf{V}_t\mathbf{\Gamma}_{ij} + \lambda_2(\mathbf{V} - \mathbf{I})), & \text{otherwise,} \end{cases} \\
b_{t+1} &= \begin{cases} b_t, & \text{if } l_{ij}\rho_{ij} \geq \alpha \\ b_t + \tau l_{ij}, & \text{otherwise,} \end{cases}
\end{aligned} \tag{6.2}$$

where τ is the learning rate, $\mathbf{\Psi}_{ij} = (x_i - x_j)(x_i - x_j)^T$, $\mathbf{\Gamma}_{ij} = x_i x_j^T + x_j x_i^T$, $\rho_{ij} = b - d_{\mathbf{W}}(x_i, x_j) + 2s_{\mathbf{V}}(x_i, x_j)$. Note that the regularization term is updated only when the condition is violated instead of being updated for every iteration. In practice, this strategy significantly reduces the computational complexity but yields similar results.

6.2.2 Metric Adaptation with Negative Set

Given a negative training set T which has no overlapping subjects with the test set and a pair of test templates G and P, we adaptively learn two metric metrics for templates G and P as described below. The positive pairs are generated by every two features in G (*i.e.*, if the template only contains a single face image, we use the features extracted from the image and its horizontally flipped one.). On the other hand, the negative pairs are generated for every two features between G and T (*i.e.*, one in G, and the other one in T.). With a bunch of positive and negative pairs, we train the regularized metric for G by solving (6.1). Once the metric matrices are learned, we compute the similarity score $\rho_G(P, G) = b_G - d_{\mathbf{W}_G}(x_G, x_P) + 2s_{\mathbf{V}_G}(x_G, x_P)$, where x_G and x_P are the average of unit-normalized features for the template (*i.e.* the average used here is media sensitive: the features from the same video will be averaged first and then averaged with others.). Similarly, we train a metric for the template P and compute $\rho_P(P, G)$. Finally, the similarity score between G and P is computed as the weighted sum of the two scores: $s(P, G) = \beta\rho_G + (1 - \beta)\rho_P$ where β is the weight used to balance the two similarity scores and is determined as the ratio of the number of positive pairs in each template. The overview of the proposed method is illustrated in Figure 6.1.

6.2.3 Negative Set Selection

In general, a large negative set is preferred for metric adaptation since more diverse negative pairs help to learn a better metric. However, since metric adaptation

is conducted during test time, it is essential to reduce the size of the negative set to speed up the computation. One simple solution is to directly average and normalize the features by subjects and use the averaged features as the negative set. However, since the training set contains some faces which may be badly aligned or in extreme pose or illumination conditions, directly averaging them with other good features introduces errors and degrades the performance. We develop a strategy to identify outliers based on the results of K-means clustering and only use the good features for averaging. First, the mean feature of each subject is used to initialize the K-means algorithm, K is set as the number of subjects in the negative set, and then we apply the K-means algorithm on the entire negative set. In the best situation, all the features should be assigned to the cluster corresponding to their ground truth labels. If some features are assigned to the clusters of other subjects, these features are potential outliers to their own subjects. Nevertheless, if the subjects contain very few features, it is possible that all the features in the subjects are assigned to other subjects. In this case, we should preserve all the features in the subjects. The detailed steps are summarized in Algorithm 3.

6.3 Experimental Results

In this section, we evaluate the proposed approach on the challenging IARPA Janus Benchmark A (IJB-A) and its extended version, the Janus Challenging Set 2 (CS2). Some alternative methods are compared and the receiver operating characteristic curves (ROC) are used to measure the performance for different algorithms.

Algorithm 3 Negative Set Selection

Input: Original Negative Set X, class labels for all the features in X.

Output: Representative negative set X_r.

1. Mean selection: For each subject i, compute the mean point x_{M_i}

2. Representative feature selection: Apply the K-means algorithm on the entire set X, using all the x_{M_i} obtained from step 3 for initialization. For each feature, compare its new cluster index with its true label. Preserve the consistent ones.

3. Outliers removing: Remove the non-consistent features. If there is no consistent feature for certain subjects, preserve all the features.

4. Representative features averaging: Average the remaining features in each subject to get the final negative set X_r.

We also discuss the reduction of model size and the selection of the negative set.

6.3.1 Experiment Setup

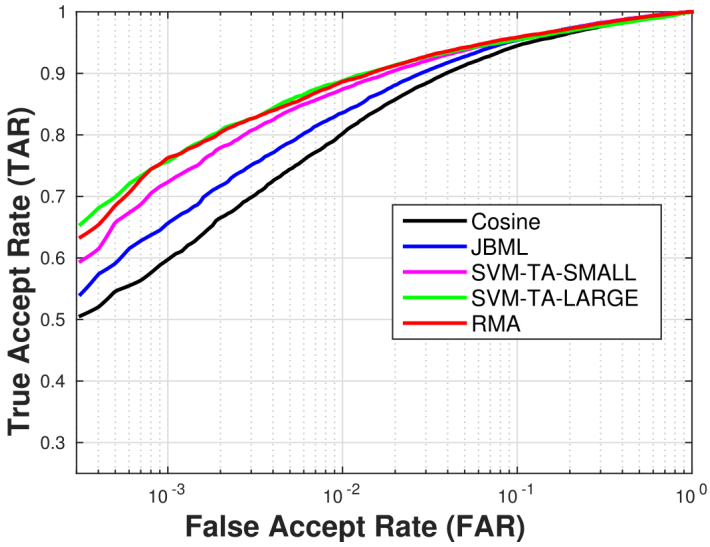
The DCNN features used in all the experiments of this work are the pool5 features extracted by the deep convolutional network proposed in [104] which consists of ten convolutional layers, five pooling layers and one fully connected layer and is trained using the CASIA-WebFace dataset [105]. The dimensionality of the pool5 features is 320. Media averaging pooling followed by unit-normalization for the feature vectors are used as the preprocessing steps after feature extraction [41].

For the parameters used in (6.1), we set margin $\alpha = 0.001$, regularization parameters $\lambda_1 = \lambda_2 = 0.01$, and the learning rate $\tau = 0.01$. In general, a large

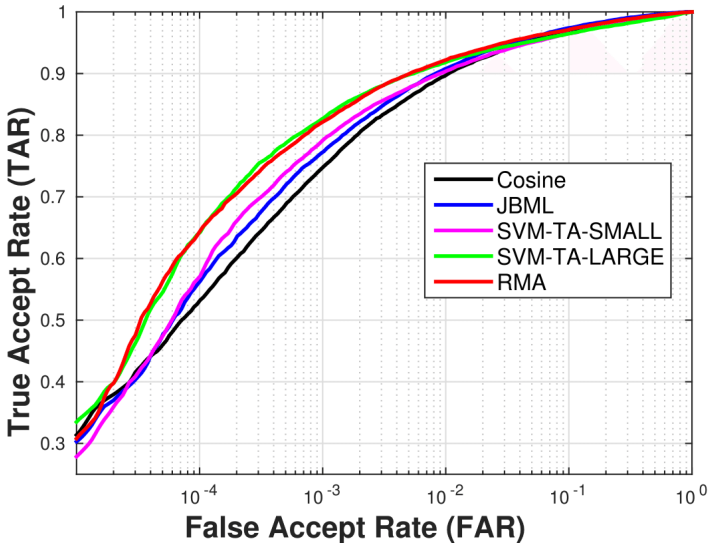
Method	Negative Set Usage, Size	TAR@FAR = 10^{-3}	TAR@FAR = 10^{-2}	TAR@FAR = 10^{-1}
Cosine	No	0.598±0.078	0.802±0.055	0.945±0.009
JBML	Yes, during training period, about 10,000	0.655±0.072	0.836±0.028	0.955±0.006
SVM-TA-v0 [41]	Yes, during adaptation period, N/A	N/A	0.939±0.013	N/A
SVM-TA-v1 [41]	Yes, during adaptation period, 332	0.723±0.034	0.874±0.012	0.956±0.006
SVM-TA-v1 [41]	Yes, during adaptation period, about 10,000	0.757±0.048	0.888±0.013	0.956±0.007
RMA	Yes, during adaptation period, 332	0.763±0.037	0.887±0.014	0.959±0.005

Table 6.1: Verification results on the IJB-A dataset. The results are averaged over 10 splits. The results of SVM-TA-v0 in the third row are directly cited from the original paper. The results of SVM-TA-v1 are implemented by us.

margin results in a more strict condition for $l_{ij}\rho_{ij} \geq \alpha$ in (6.2), where the condition is easier to be violated and the metric will be updated very often. This may discourage the metric from learning the hard positives or negatives. Therefore, we set the margin to a relatively small number so that the metric is updated based on the hard negative/positive pairs. This idea is similar to the hard negative/positive mining strategy which is widely used in metric learning and has proven to be effective [45, 106, 107]. The learning rate and the regularization parameter are determined based on cross validation. We initialize $\mathbf{W_0} = \mathbf{V_0} = \mathbf{I}$ and b_0 is learned using only the negative set during the training period. The size of negative set is 332 which is the number of subjects in the set. In our experiments, all the possible positive and negative pairs are used to learn the metric for five epochs because the size of the negative set and the test templates are small. The weight used to balance the two similarity scores is set as the ratio of the number of positive pairs in each template.



(a)



(b)

Figure 6.2: ROC curves for IJB-A and CS2 dataset. The results are averaged over 10 splits. SVM-TA-SMALL means using a small negative set and SVM-TA-LARGE means using a large negative set where SVM-TA refers to our implementation, SVM-TA-v1.

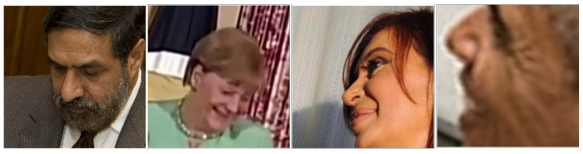


Figure 6.3: Sample images in IJB-A dataset.

6.3.2 Evaluation on IJB-A and CS2 Datasets

Both IJB-A and JANUS CS2 datasets contain 500 subjects with 5,397 images and 2,042 videos. The datasets are divided into training sets which contain 333 subjects, and test sets which contain 167 subjects. Based on the different training/ test set division, ten splits are generated. Some sample images are shown in Figure 6.3. The training sets are shared for both datasets. For the test set, JANUS CS2 contains about 167 gallery templates and 1763 probe templates. All pairs of gallery-to-probe templates are used for verification. The IJB-A evaluation protocol selects around 11,748 hard pairs of gallery-to-probe templates (1,756 positive and 9,992 negative pairs) from JANUS CS2.

We compare the results of the proposed regularized metric adaptation (RMA) approach with two baseline methods, the cosine similarity without metric learning and the joint Bayesian metric learning (JBML) without metric adaptation. The cosine similarity method is unsupervised and does not require any training set while JBML is trained using the training data of IJB-A and JANUS CS2 during the training period and the trained model is then applied in the test phase. We also compare our results with the recently proposed SVM-based template adaptation

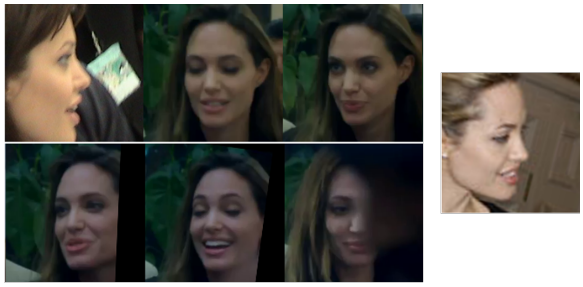


Figure 6.4: Sample pair that is correctly classified by RMA while mis-classified by JBML.

(SVM-TA) method [41], which requires a large negative set in test phase for template adaptation. We cite the results from [41] as SVM-TA-v0. We also follow the same preprocessing steps and use the same parameters described in [41] for our implemented features as SVM-TA-v1 for comparison. The main difference comes from the DCNN features used in both works where in [41] the network is trained using the VGG face dataset which contains more face images (around 2.6 million faces) than the CASIA-WebFace dataset (around 500K faces) used by us.

Method	Negative Set Usage, Size	TAR@FAR = 10^{-3}	TAR@FAR = 10^{-2}	TAR@FAR = 10^{-1}
Cosine	No	0.748±0.031	0.898 ±0.010	0.945 ±0.003
JBML	Yes, during training period, about 10,000	0.773±0.040	0.908±0.007	0.974±0.004
SVM-TA-v1 [41]	Yes, during adaptation period, 332	0.792±0.018	0.904±0.007	0.965±0.004
SVM-TA-v1 [41]	Yes, during adaptation period, about 10,000	0.827±0.014	0.918± 0.007	0.965±0.003
RMA	Yes, during adaptation period, 332	0.822±0.019	0.922±0.008	0.971±0.002

Table 6.2: Verification results on CS2 dataset. The results are averaged over 10 splits.

Figure 6.2(a) shows the ROC curves for the IJB-A dataset. Table 6.1 shows the TAR for FARs at $10^{-3}, 10^{-2}, 10^{-1}$. The results are averaged over 10 splits. It is observed that the proposed method RMA shows better results than other non-adaptation baselines especially in the low FAR region. Figure 6.4 shows an example pair that is correctly classified by RMA, yet mis-classified by JBML at $FAR = 10^{-2}$. It demonstrates the effectiveness of the metric adaptation approach for the hard case, where extreme poses and occlusions are present. Notice that two versions of the SVM-TA-v1 results are reported based on whether a small or a large negative set is used. We outperform SVM-TA-v1 when using the same negative set while perform comparably when SVM-TA-v1 uses a larger negative set. It demonstrates that metric learning can fully exploit the discriminative information in a relatively small negative set.

Figure 6.2(b) shows the ROC curves for the CS2 dataset. Table 6.2 shows the performance of different methods on the CS2 dataset. Results are averaged over 10 splits. As an extended version of IJB-A dataset, the CS2 dataset compares all the possible pairs in the gallery and probe sets. The baseline for CS2 is higher than for the IJB-A dataset which makes it more difficult to improve from the baseline. The proposed RMA still outperforms the non-adaptation method by 2% at $FAR = 10^{-2}$ and 5% at $FAR = 10^{-3}$. SVM-TA-v1 with the large negative set still yields comparable results. However, when using the same negative set, it can hardly improve the performance from the non-adaptation baselines.

6.3.3 Model Size Reduction

When the model learned by the metric adaptation needs to be saved for future use (*e.g.*, the subject is enrolled in the database.), it is useful to reduce the model size as small as possible for practical use. The original model requires $\mathcal{O}(n^2)$ storage space where n is the dimension of the data sample. Since the model is template-specific, the whole model size for a dataset will be proportional to the number of unique templates which is usually very large. We reduce the original model size to $\mathcal{O}(n)$ by taking only the diagonal of $\mathbf{W}$ and the transformed feature $\mathbf{V}^T\mathbf{V}x$ for each template. The similarity is then computed as $\rho_G(x_G, x_P) = b_G - (x_G - x_P)^T diag(\mathbf{W}_G)^2(x_G - x_P) + 2x_P\mathbf{V}_G^T\mathbf{V}_G x_G$ and similarly for $\rho_P(x_G, x_P)$. The reason why we keep the diagonal elements of $\mathbf{W}$ is that as we enforce a regularization term in (6.1), which guarantees that the elements on the diagonal preserve the most information as compared to other off-diagonal elements. The results with and without model size reduction are listed in Table 6.3. From the table, the performance only decreases by a small margin while the whole model size is significantly reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$.

Model Size	TAR@FAR = 10^{-3}	TAR@FAR = 10^{-2}	TAR@FAR = 10^{-1}
$\mathcal{O}(n)$	0.746±0.041	0.878±0.016	0.956±0.005
$\mathcal{O}(n^2)$	0.763±0.037	0.887±0.014	0.959±0.005

Table 6.3: The results for the model size reduction which are averaged over 10 splits.

6.3.4 Negative Set Selection Analysis

The size of the negative set significantly influences the adaptation time as well as the storage space. It is desired to keep a relatively small negative set while maintaining a similar performance as the large one. We investigate and compare different strategies to reduce the size, including (1) ***Random*** where a media feature (*i.e.* features from the same media are averaged) for each subject is randomly selected into the negative set, (2) ***Naive K-means*** where the media average feature for each subject (*i.e.* features from the same media are averaged first and then different media from one subject are averaged) is used as the negative set, (3) ***Naive K-medoids*** where the 1-medoid of all the media features of each subject is taken into the negative set, (4) ***Outlier Removed K-means*** means the method described in Algorithm 3, and (5) ***Outlier Removed K-medoids*** means the similar strategy described as *Outlier Removed K-means* but K-means is replaced by K-medoids.

Table 6.4 summarizes the results of different methods using RMA on IJB-A verification split 1. It shows that methods based on K-means outperform K-medoids based method and randomly selection by a large margin. It shows that by averaging different media in one subject, we obtain more discriminative information than just including a single media feature. The *Outlier Removed Kmeans* performs slightly better than *Naive Kmeans* at $FAR = 10^{-2}$.

Method	TAR@FAR = 10^{-3}	TAR@FAR = 10^{-2}	TAR@FAR = 10^{-1}
Random	0.683	0.848	0.943
Naive K-means	0.773	0.886	0.952
Outlier Removed K-means	0.770	0.890	0.953
Naive K-medoids	0.672	0.851	0.946
Outlier Removed K-medoids	0.673	0.851	0.947

Table 6.4: Negative set selection. It shows the results of different strategies for the split 1 of the IJB-A face verfication.

6.4 Conclusion

In this chapter, we proposed a regularized metric adaptation approach to learn a template-specific metric for the set-based face verification problem. Extensive experiments on the newly released IARPA Janus Benchmark A(IJB-A) and CS2 dataset demonstrate the effectiveness of the proposed method for unconstrained face verification when the negative set is used. In addition, the proposed approach can be used to significantly reduce the model size while still yielding comparable performance to the original model. Analysis shows the importance of the negative set selection on the verification performance. A K-means based method can efficiently construct a compact and representative negative set.

Chapter 7: Conclusions and Directions for Future Research

7.1 Conclusions

In this dissertation, we begin with studying the effects of covariates on unconstrained face verification. Our evaluations are based on deep learning networks and large training data sets. We find that most studied covariates remarkably affect the face verification performance. Pose variations and occlusions are the top confounding factors that cause performance drop by large margins. Indoor is more favorable than outdoors for image acquisition. In addition, different demographic groups present significant differences on performance. Males are easier to verify than females and old subjects generally performs better than young ones. For skin tone, light pink achieves the best performance while medium-dark brown performs the worst.

Based on experimental results, we proposed several domain adaptation methods to mitigate the negative effects of these covariates. In Chapter 3, we showed the benefit of cooperating with the pose verification task for pose-robust face verification. We proposed a joint model to learn the metrics for the two tasks together and enforce an orthogonal regularization on the learned projection matrices for the two tasks. By excluding the information contained in the auxiliary task, the learned

metric for face verification is more pose-robust. We conducted extensive experiments on three challenging datasets and the experimental results show that the proposed approach improves the baseline methods.

In Chapter 4 and Chapter 5, we presented two methods to improve blurred face recognition performance. In Chapter 4, we applied an incremental dictionary learning method to explicitly reduce the domain mismatch. We utilized the supportive samples to smoothly connect the source and target domains and designed an efficient stopping criterion to guarantee the adaptation reduces the domain dissimilarity monotonically. We also proposed an unsupervised face deblurring method to restore the latent sharp images in Chapter 5. We utilized the idea of disentangled representation to split the content and blur features in a blurred image. By adding KL divergence loss, the blur features are discouraged to encode content information. In order to preserve the content structure of the original images, we added a blurring branch and cycle-consistency loss to the framework. The perceptual loss helps the blurred image remove unrealistic artifacts.

In Chapter 6, we proposed a template adaptation approach to ensure that the metric learned by training set can generalize well to test data. Template-specific metrics was learned by using each test templates and the negative sets. A regularizer was added to efficiently reduce the model size while still yielding comparable performance to the original model.

7.2 Directions for Future Research

In Chapter 2, we studied the effects of covariates. Some of the results from our studies show promising research directions. First, apart from the yaw problem, we should also consider the influence of roll when designing face verification systems. This can be done by either improved face alignment or more robust feature extraction models. Second, since gender, age and skin tone all have significant impact on performance, we may collect the training set more carefully to improve the performance on certain demographic groups. Third, just as gender estimation was helpful for data curation, other covariates like race may also be used in a similar way.

In Chapter 3, we developed a metric learning approach for pose-robust face verification. To extend the proposed multi-task framework, we could develop a method for training pair selection. In the experiments, we found the selection of training pairs to be crucial for improving verification performance. This is because the discriminative capability of the learned metric is affected by the spread of the training data. Moreover, since the features used for both tasks are extracted from the same feature pool, we also need to simultaneously consider the diversity of the pose distribution of the training data. Another possible research direction is to explore more auxiliary tasks. Tasks like age verification and expression verification are also competing tasks with respect to face verification. We could also add orthogonal constraints to the metric learned for these tasks and to the metric for face verification.

In Chapter 4, incremental dictionary learning method was used to reduce

domain mismatch. A possible direction to extend the proposed method is to develop online methods. Most existing domain adaptive methods assume that source and target domains are static. However, in practice target domains are usually dynamic and evolving over time. For example, in surveillance videos, the light condition changes gradually from day to night and the background may vary due to the weather change. Another simple example could be the aging problem for face datasets. Gallery faces are captured at one time and probe faces become older over time. In these cases, only performing adaptation once cannot meet the requirements. The model needs to be updated dynamically when new probe data is acquired.

In Chapter 5, an unsupervised domain-specific deblurring method was proposed to restore latent sharp images. A straightforward extension would be designing a generic method for natural image deblurring. In a priliminary experiment, we find directly applying our methods to generic images does not work well. Colors may change and details are missing in the deblurred results. Another promising direction is to explore this idea to other tasks like dehazing and super-resolution.

Bibliography

[1] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[2] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014.

[3] Yueqi Duan, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Context-aware local binary feature learning for face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1139–1153, 2018.

[4] Jiwen Lu, Gang Wang, and Jie Zhou. Simultaneous feature and dictionary learning for image set based face recognition. *IEEE Transactions on Image Processing*, 26(8):4042–4054, 2017.

[5] Jiwen Lu, Junlin Hu, and Jie Zhou. Deep metric learning for visual understanding: An overview of recent advances. *IEEE Signal Processing Magazine*, 34(6):76–84, 2017.

[6] P Jonathon Phillips, Amy N Yates, Ying Hu, Carina A Hahn, Eilidh Noyes, Kelsey Jackson, Jacqueline G Cavazos, Géraldine Jeckeln, Rajeev Ranjan, Swami Sankaranarayanan, Jun-Cheng Chen, Carlos D. Castillo, Rama Chellappa, David White, and Alice J. OToole. Face recognition accuracy of forensic examiners, superrecognizers, and face recognition algorithms. *Proceedings of the National Academy of Sciences*, 115(24):6171–6176, 2018.

[7] Connor J Parde, Carlos Castillo, Matthew Q Hill, Y Ivette Colon, Swami Sankaranarayanan, Jun-Cheng Chen, and Alice J OToole. Face and image representation in deep CNN features. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 673–680. IEEE, 2017.

[8] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2892–2900, 2015.

[9] Boyu Lu, Jingxiao Zheng, Jun-Cheng Chen, and Rama Chellappa. Pose-robust face verification by exploiting competing tasks. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 1124–1132. IEEE, 2017.

[10] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017.

[11] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[12] Zongyi Liu and Sudeep Sarkar. Outdoor recognition at a distance by fusing gait and face. *Image and Vision Computing*, 25(6):817–832, 2007.

[13] Yui Man Lui, David Bolme, Bruce A Draper, J Ross Beveridge, Geoff Givens, and P Jonathon Phillips. A meta-analysis of face recognition covariates. In *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*, pages 1–8. IEEE, 2009.

[14] J Ross Beveridge, Geof H Givens, P Jonathon Phillips, and Bruce A Draper. Factors that influence algorithm performance in the face recognition grand challenge. *Computer Vision and Image Understanding*, 113(6):750–762, 2009.

[15] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In *European Conference on Computer Vision*. Springer, 2016.

[16] Wei-An Lin, Jun-Cheng Chen, and Rama Chellappa. A proximity-aware hierarchical clustering of faces. *arXiv preprint arXiv:1703.04835*, 2017.

[17] Changxing Ding and Dacheng Tao. A comprehensive survey on pose-invariant face recognition. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(3):37, 2016.

[18] Haoxiang Li, Gang Hua, Zhe Lin, Jonathan Brandt, and Jianchao Yang. Probabilistic elastic matching for pose variant face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3499–3506, 2013.

[19] Zhenyao Zhu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning identity-preserving face space. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 113–120, 2013.

[20] Meina Kan, Shiguang Shan, Haihong Zhang, Shihong Lao, and Xilin Chen. Multi-view discriminant analysis. In *European Conference on Computer Vision*, pages 808–821. Springer, 2012.

[21] Changxing Ding, Chang Xu, and Dacheng Tao. Multi-task pose-invariant face recognition. *IEEE Transactions on Image Processing*, 24(3):980–993, 2015.

[22] Wael AbdAlmageed, Yue Wu, Stephen Rawls, Shai Harel, Tal Hassner, Iacopo Masi, Jongmoo Choi, Jatuporn Lekust, Jungyeon Kim, Prem Natarajan, et al. Face recognition using deep multi-pose representations. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[23] Boyu Lu, Jun-Cheng Chen, Carlos D Castillo, and Rama Chellappa. An experimental evaluation of covariates effects on unconstrained face verification. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2019.

[24] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1628–1636, 2016.

[25] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1107–1114, 2013.

[26] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[27] Ziyi Shen, Wei-Sheng Lai, Tingfa Xu, Jan Kautz, and Ming-Hsuan Yang. Deep semantic face deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[28] Michal Hradi, Jan Kotera, Pavel Zemk, and Filip roubek. Convolutional neural networks for direct text deblurring. In Mark W. Jones Xianghua Xie and Gary K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 6.1–6.13. BMVA Press, September 2015.

[29] Zhe Hu, Sunghyun Cho, Jue Wang, and Ming-Hsuan Yang. Deblurring low-light images with light streaks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3382–3389, 2014.

[30] Jinshan Pan, Zhe Hu, Zhixun Su, and Ming-Hsuan Yang. Deblurring text images via l0-regularized intensity and gradient prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2901–2908, 2014.

[31] Jinshan Pan, Zhe Hu, Zhixun Su, and Ming-Hsuan Yang. Deblurring face images with exemplars. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 47–62. Springer, 2014.

[32] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiri Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[33] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3883–3891, 2017.

[34] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 769–777, 2015.

[35] Ayan Chakrabarti. A neural approach to blind motion deblurring. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 221–235. Springer, 2016.

[36] L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In *Asian Conference on Computer Vision*, pages 88–97. 2010.

[37] Huimin Guo, William Robson Schwartz, and Larry S Davis. Face verification using large feature sets and one shot similarity. In *International Joint Conference on Biometrics*, pages 1–8. IEEE, 2011.

[38] S. Xie, S. G. Shan, X. L. Chen, and J. Chen. Fusing local patterns of gabor magnitude and phase for face recognition. *IEEE Transactions on Image Processing*, 19(5):1349–1361, 2010.

[39] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.

[40] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893. IEEE, 2005.

[41] N. Crosswhite, J. Byrne, O. M. Parkhi, C. Stauffer, Q. Cao, and A. Zisserman. Template Adaptation for Face Verification and Identification. *ArXiv e-prints*, March 2016.

[42] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K. Jain, James A. Duncan, Kristen

Allen, Jordan Cheney, and Patrick Grother. Iarpa janus benchmark-b face dataset. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[43] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney, et al. Iarpa janus benchmark–c: Face dataset and protocol. In *Proceedings of the IAPR International Conference on Biometrics*, 2018.

[44] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D Castillo, and Rama Chellappa. An all-in-one convolutional neural network for face analysis. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 17–24. IEEE, 2017.

[45] Swami Sankaranarayanan, Azadeh Alavi, and Rama Chellappa. Triplet similarity embedding for face verification. *arXiv preprint arXiv:1602.03418*, 2016.

[46] Ankan Bansal, Anirudh Nanduri, Carlos Castillo, Rajeev Ranjan, and Rama Chellappa. Umdfaces: An annotated face dataset for training deep networks. *arXiv preprint arXiv:1611.01484*, 2016.

[47] Ankan Bansal, Carlos Castillo, Rajeev Ranjan, and Rama Chellappa. The do's and don'ts for CNN-based face verification. *arXiv preprint arXiv:1705.07426*, 2017.

[48] Aaron Nech and Ira Kemelmacher-Shlizerman. Level playing field for million scale face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[49] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515. Springer, 2016.

[50] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[51] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284. AAAI Press, 2017.

[52] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[53] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[54] P JONATHON Phillips, Patrick Grother, ROSS J Micheals, DUANE M Blackburn, ELHAM Tabassi, and MIKE BONE. Evaluation report. 2003.

[55] J Ross Beveridge, Geof H Givens, P Jonathon Phillips, Bruce A Draper, David S Bolme, and Yui Man Lui. Frvt 2006: Quo vadis face quality. *Image and Vision Computing*, 28(5):732–743, 2010.

[56] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[57] Huy Tho Ho and Rama Chellappa. Pose-invariant face recognition using markov random fields. *IEEE transactions on image processing*, 22(4):1573–1584, 2013.

[58] Dong Chen, Xudong Cao, Liwei Wang, Fang Wen, and Jian Sun. Bayesian face revisited: A joint formulation. In *Computer Vision–ECCV 2012*, pages 566–579. Springer, 2012.

[59] Jun-Cheng Chen, Vishal M Patel, and Rama Chellappa. Unconstrained face verification using deep CNN features. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[60] Binod Bhattarai, Gaurav Sharma, and Frederic Jurie. Cp-mtml: Coupled projection multi-task metric learning for large scale face retrieval. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[61] Brendan F Klare, Ben Klein, Emma Taborsky, Austin Blanton, Jordan Cheney, Kristen Allen, Patrick Grother, Alan Mah, Mark Burge, and Anil K Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1931–1939. IEEE, 2015.

[62] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

[63] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *arXiv preprint arXiv:1603.01249*, 2016.

[64] Lorenzo Bruzzone and Mattia Marconcini. Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(5):770–787, 2010.

[65] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 999–1006. IEEE, 2011.

[66] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073. IEEE, 2012.

[67] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2960–2967. IEEE, 2013.

[68] Mahsa Baktashmotlagh, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 769–776. IEEE, 2013.

[69] Boqing Gong, Kristen Grauman, and Fei Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *Proceedings of The 30th International Conference on Machine Learning*, pages 222–230, 2013.

[70] Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608, 2006.

[71] Tatiana Tommasi and Barbara Caputo. Frustratingly easy nbnn domain adaptation. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 897–904. IEEE, 2013.

[72] Amaury Habrard, Jean-Philippe Peyrache, and Marc Sebban. Iterative self-labeling domain adaptation for linear structured image classification. *International Journal on Artificial Intelligence Tools*, 22(05), 2013.

[73] Chun-Wei Seah, Yew-Soon Ong, and Ivor W Tsang. Combating negative transfer from predictive distribution differences. *Cybernetics, IEEE Transactions on*, 43(4):1153–1165, 2013.

[74] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696. ACM, 2009.

[75] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[76] Jie Ni, Qiang Qiu, and Rama Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 692–699. IEEE, 2013.

[77] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010*, pages 213–226. Springer, 2010.

[78] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

[79] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010.

[80] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer joint matching for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1410–1417. IEEE, 2014.

[81] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of The 31st International Conference on Machine Learning*, pages 647–655, 2014.

[82] D Zhang, Meng Yang, and Xiangchu Feng. Sparse representation or collaborative representation: Which helps face recognition? In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 471–478. IEEE, 2011.

[83] Yi-Chen Chen, Vishal M Patel, Jaishanker K Pillai, Rama Chellappa, and P Jonathon Phillips. Dictionary learning from ambiguously labeled data. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 353–360. IEEE, 2013.

[84] Boyu Lu, Rama Chellappa, and Nasser M. Nasrabadi. Incremental dictionary learning for unsupervised domain adaptation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2015.

[85] Thekke Madam Nimisha, Kumar Sunil, and AN Rajagopalan. Unsupervised class-specific deblurring. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 221–235. Springer, 2018.

[86] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[87] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 2868–2876. IEEE, 2017.

[88] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Towards open-set identity preserving face synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6713–6722, 2018.

[89] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.

[90] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 1520–1529. IEEE, 2017.

[91] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[92] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[93] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[94] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 36–52. Springer, 2018.

[95] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[96] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[97] G Boracchi and A Foi. Modeling the performance of image restoration from motion blur. *IEEE Transactions on Image Processing*, 21(8):3502–3517, 2012.

[98] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[99] S Sengupta, JunCheng Cheng, C.D Castillo, V.M Patel, R Chellappa, and D.W Jacobs. Frontal to profile face verification in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, February 2016.

[100] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2015.

[101] Dilip Krishnan, Terence Tay, and Rob Fergus. Blind deconvolution using a normalized sparsity measure. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 233–240. IEEE, 2011.

[102] Boyu Lu, Jun-Cheng Chen, and Rama Chellappa. Regularized metric adaptation for unconstrained face verification. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 4112–4117. IEEE, 2016.

[103] Wei-Sheng Lai, Jia-Bin Huang, Zhe Hu, Narendra Ahuja, and Ming-Hsuan Yang. A comparative study for single image blind deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 1701–1709, 2016.

[104] Jun-Cheng Chen, Vishal M Patel, and Rama Chellappa. Unconstrained face verification using deep CNN features. *arXiv preprint arXiv:1508.01722*, 2015.

[105] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

[106] Hailin Shi, Xiangyu Zhu, Shengcai Liao, Zhen Lei, Yang Yang, and Stan Z Li. Constrained deep metric learning for person re-identification. *arXiv preprint arXiv:1511.07545*, 2015.

[107] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.