ABSTRACT

Title of dissertation: ROBUST REPRESENTATIONS FOR UNCONSTRAINED FACE RECOGNITION AND ITS APPLICATIONS

Jun-Cheng Chen, Doctor of Philosophy, 2016

Dissertation directed by: Professor Rama Chellappa
Department of Computer Science

Face identification and verification are important problems in computer vision and have been actively researched for over two decades. There are several applications including mobile authentication, visual surveillance, social network analysis, and video content analysis. Many algorithms have shown to work well on images collected in controlled settings. However, the performance of these algorithms often degrades significantly on images that have large variations in pose, illumination and expression as well as due to aging, cosmetics, and occlusion.

How to extract robust and discriminative feature representations from face images/videos is an important problem to achieve good performance in uncontrolled settings. In this dissertation, we present several approaches to extract robust feature representation from a set of images/video frames for face identification and verification problems.

We first present a dictionary approach with dense facial landmark features. Each face video is segmented into K partitions first, and the multi-scale features are extracted

from patches centered at detected facial landmarks. Then, compact and representative dictionaries are learned from dense features for each partition of a video and then concatenated together into a video dictionary representation for the video. Experiments show that the representation is effective for the unconstrained video-based face identification task. Secondly, we present a landmark-based Fisher vector approach for video-based face verification problems. This approach encodes over-complete local features into a high-dimensional feature representation followed by a learned joint Bayesian metric to project the feature vector into a low-dimensional space and to compute the similarity score. We then present an automated system for face verification which exploits features from deep convolutional neural networks (DCNN) trained using the CASIA-WebFace dataset. Our experimental results show that the DCNN model is able to characterize the face variations from the large-scale source face dataset and generalizes well to another smaller one. Finally, we also demonstrate that the model pre-trained for face identification and verification tasks encodes rich face information which benefit other face-related tasks with scarce annotated training data. We use apparent age estimation as an example and develop a cascade convolutional neural network framework which consists of age group classification and age regression, and a deep networks is fine-tuned using the target data.

ROBUST REPRESENTATIONS FOR UNCONSTRAINED FACE RECOGNITION AND ITS APPLICATIONS

by

Jun-Cheng Chen

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
2016

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor Larry S. Davis,
Professor David W. Jacobs
Professor Ramani Duraiswami
Professor Tom Goldstein

Dedication

To my parents, Yeon-Ru Kuo and Chin-Ron Chen, and my sister, Jizhen Chen.

Acknowledgments

First, I sincerely thank my advisor, Professor Rama Chellappa for giving a chance working with him, and guiding me to tackle challenging and interesting problems in computer vision over the past years. The discussions with him were always encouraging and inspiring. His wisdom for leadership, his dedication to work, and positive attitude towards life, and great sense of humor are the sources of inspiration to me for my research.

Besides my advisor, I would like to thank the rest of my dissertation committee: Prof. Larry Davis, Prof. David Jacobs, Prof. Ramani Duraiswami, and Prof. Tom Goldstein, for their valuable feedbacks and suggestions on this dissertation which inspire me to further explore my research from various perspectives.

My graduate life has been enriched in many ways by fellow colleagues at the Computer Vision Lab, among whom I would like to particularly thank Dr. Vishal Patel, Dr. Ming-Yu Liu, Dr. Kaushik Mitra, Dr. Qiu Qiang, Dr. Aswin Sankaranarayanan, Dr. Carlos Castillo, Dr. Ruiping Wang, Ching-Hui Chen, Jingxiao Zheng, Boyu Lu, Wei-An Lin, Dr. Jingjing Zheng, Dr. Jie Ni, Dr. Ming Du, Rajeev Ranjan, Amit Kumar, Swaminathan Sankaranrayanan for their fruitful discussion and collaboration on research projects.

In addition, many friends have helped me overcome setbacks and stay focused on my research through these difficult years. I would like to acknowledge Hsueh-Chien Cheng, Cheng-Chih Yang, Dr. Xi Chen, Dr. Qi Hu who was always there to listen and gave me spiritual support.

I also thank Jennifer Story, Fatima Bangura, Janice Perrone, Melanie Prange and Arlene Schenk for all the administrative help.

Finally, I would like to thank my family: my parents and my sister. Without their support, I am unable to make it.

There are much more people who help me during the period of my PhD study. I apologize to those I have inadvertently left out.

Table of Contents

List of Figures

Chapter 1: Introduction

1.1 Motivation

Face recognition is one of the active research areas in computer vision and has a wide range of practical applications including surveillance, social network, and mobile authentication [4]. Even though many face recognition algorithms have shown promising results in controlled settings, unconstrained face recognition is still a challenging problem due to large variations in pose, lighting, blur, expression and occlusion. Therefore, how to extract robust and discriminative representation from face images/videos is an important problem. In this dissertation, we present several approaches to extract robust feature representation from a set of images/video frames for face recognition problems. In general, face recognition can be broadly classified into two major tasks: identification and verification. We focus on face identification and verification problems in this dissertation. (*i.e.* the purpose of the face identification problem is to determine the subject identity from the given candidate set, and the face verification problem is to determine whether two face images belong to the same person or not.)

1.2 Dictionary-based Video Face Recognition Using Dense Multi-scale Facial Landmark Features

To handle large face variations in unconstrained settings, many methods have been proposed to learn an invariant and discriminative representation from face images and videos. Coates *et al.* [5] showed that an over-complete representation is critical for achieving high recognition rates regardless of the encoding methods. In [6], it was shown that densely sampling overlapped image patches helps to improve the recognition performance. In the first part of the dissertation, we propose a dictionary-based approach using dense and high-dimensional features extracted from multi-scale patches centered at detected facial landmarks for video-to-video face identification problem. The idea is to utilize dictionary learning technique to learn a compact video representation from discriminative high-dimensional dense landmark features extracted from each frame of a video. Subsequently, dictionary learning is applied to each image set and video independently without requiring any extra training data. This approach improves the recognition performance compared with image-set based recognition approach.

1.3 Landmark-based Fisher Vector Representation for Video-based Face Verification

For the face verification problem, one usually measures the performance using the receiver operating characteristic curves (ROC) which is generated based on the ranked similarity scores from all of the matched and non-matched face pairs. Therefore, besides

the robust face representation, learning a discriminative distance measure is the other key component for boosting the performance. In the second part of the dissertation, we present an approach based on Fisher vector representation (FV) for the face verification problem. We first extract over-complete local features from patches around facial landmarks and encodes them using FV into a high-dimensional feature followed by a learned joint Bayesian metric to project the feature vector into a low-dimensional space and compute the similarity score. Our approach achieves good results on the Point and Shoot Challenge dataset (PaSC) [7] dataset compared to other methods reported in IJCB 2014 face recognition competition.

1.4 Unconstrained Still/Video-Based Face Verification with Deep Convolutional Neural Networks

In the third part of the dissertation, since deep convolutional neural networks (DCNN) have demonstrated top performances on different computer vision tasks, including object recognition [8] [9], object detection [10], and face verification [11]. In contrast to approaches based on high-dimensional feature representation, it has been shown that a DCNN model can not only characterize large data variations but also learn a compact and discriminative feature representation when the size of the training data is sufficiently large. Once the model is learned, it is possible to generalize it to other tasks by fine-tuning the learned model on target datasets [12]. We also train a DCNN model using a comparatively small-scale face dataset - the CASIA-WebFace [13], and compare the performance of our method with other commercial off-the-shelf face matchers on the new challeng-

ing IJB-A dataset which contains full variations in pose, illumination, aging, expression, resolution and occlusion.

1.5 A Cascaded Convolutional Neural Network for Age Estimation of Unconstrained Faces

Since the pre-trained face DCNN model encodes rich information about faces, we utilize it to address other face-related tasks for which large-scale annotated datasets are not readily available. As an example, we consider the task of facial age estimation. We show that after fine-tuning the DCNN model pre-trained on the CASIA-WebFace to age estimation task, we could get reasonable performance. In addition, based on the fine-tuning technique, we propose a coarse-to-fine approach for estimating the facial age from unconstrained face images. The method consists of three modules. The first one is a DCNN-based age group classifier which classifies a given face image into age groups. The second module is a collection of DCNN-based regressors which compute the fine-grained age estimate corresponding to each age class. Finally, any erroneous age prediction is corrected using an error-correcting mechanism. Experimental evaluations on three publicly available datasets for age estimation show that the proposed approach is able to reliably estimate the age; in addition, the coarse-to-fine strategy and the error correction module significantly improve the performance.

1.6 Contributions

In this dissertation, we make the following contributions:

1. We have extensively studied the problem of robust representation for the unconstrained face verification problem. We evaluate different approaches from dictionary learning, Fisher vector to deep learning for the unconstrained face verification problem.

2. We develop an automated system for still/video-based face verification which directly takes images or videos as input and computes the similarity scores and yield robust performance to pose, illumination, and other variations.

3. We adapt the face identification/verification deep network to other face-related applications, such as facial age estimation.

1.7 Organization

The dissertation is organized as follows. In Chapter 2, we briefly review relevant related works in the literature. In Chapter 3, we present a dictionary-based approach using dense high-dimensional feature extracted from the patches around facial landmarks for unconstrained video-to-video face identification problems. In Chapter 4, we propose a landmark-based Fisher vector representation for video-based face verification. In Chapter 5, we present an automatic face verification system for unconstrained face verification using deep convolutional neural networks learned from a large-scale face dataset. In Chapter 6, we present an age estimation approach which finetunes the pre-trained DCNN model on the face identification to perform age group classification and age regression. The cascade DCNN model of both age group classification and regression demonstrate good results. In Chapter 7, we conclude and discuss future research directions.

Chapter 2: Related Work

Due to a large amount of related works for robust representations to face verification and face-related application, we briefly review them as follows. In addition, we also go through relevant works for the face preprocessing which is also important to a face verification system or other face-related applications.

2.1 Face Preprocessing

A typical face verification system consists of the following components: (1) face detection and (2) face association across video frames, (3) facial landmark detection to align faces, and (4) face verification to verify the identity of a subject. In the following subsections, we briefly discuss the preprocessing modules.

2.1.1 Face Detection

The face detection method introduced by Viola and Jones [14] is based on cascaded classifiers built using the Haar wavelet features. Since then, a variety of sophisticated cascade-based face detectors such as Joint Cascade [15], SURF Cascade [16] and CascadeCNN [17] have demonstrated improved performance. Zhu *et al.* [18] improved the performance of face detection algorithm using the deformable part model (DPM) ap-

proach, which treats each facial landmark as a part and uses the HOG features to simultaneously perform face detection, pose estimation, and landmark localization. A recent face detector, Headhunter [19], shows competitive performance using a simple DPM. However, the key challenge in unconstrained face detection is that features like Haar wavelets and HOG do not capture the salient facial information at different poses and illumination conditions. To overcome these limitations, few deep CNN-based face detection methods have been proposed in the literature such as Faceness [20], DDFD [21] and CascadeCNN [17]. It has been shown in [12] that a deep CNN pre-trained with the Imagenet dataset can be used as a meaningful feature extractor for various vision tasks. The method based on Regions with CNN (R-CNN) [22] computes region-based deep features and attains state-of-art face detection performance. In addition, since the deep pyramid [23] removes the fixed-scale input dependency in deep CNNs, it is attractive to be integrated with the DPM approach to further improve the detection accuracy across scale [24]. Ranjan *et al.* [25] proposed a multi-task face detector based on R-CNN which simultaneously detects fiducial points, head pose, face bounding boxes and gender.

2.1.2 Facial Landmark Detection

Facial landmark detection is an important component for a face verification system to align faces into canonical coordinates and to improve the performance of verification algorithms. Pioneering works such as Active Appearance Models (AAM) [26] and Active Shape Models (ASM) [27] are built using the PCA constraints on appearance and shape. In [28], Cristinacce *et al.* generalized the ASM model to a Constrained Local

Model (CLM), in which every landmark has a shape constrained descriptor to capture the appearance. Zhu *et al.* [18] used a part-based model for face detection, pose estimation and landmark localization assuming the face shape to be a tree structure. Asthana *et al.* [29] combined the discriminative response map fitting with CLM. In addition, Cao *et. al.* [30] followed the cascaded pose regression (CPR) proposed by $Doll\acute{a}r$ *et. al.* [31]: feature extraction followed by a regression stage. However unlike CPR which uses pixel difference as features, it trains a random forest based on local binary patterns. In general, these methods learn a model that directly maps the image appearance to the target output. Nevertheless, the performance of these methods depends on the robustness of local descriptors. In [8], the deep features are shown to be robust to different challenging variations. Sun *et al.* [32] proposed a cascade of carefully designed CNNs, in which at each level, outputs of multiple networks are fused for landmark estimation and achieve good performance. Unlike [32], Kumar *et al.* [1] uses a single CNN, carefully designed to provide a unique key-point descriptor and achieve better performance. In addition, Ranjan *et al.* [25] proposed a multi-task face detector based on R-CNN which simultaneously detects fiducial points, head pose, face bounding boxes and gender

2.1.3 Face Association

The video-based face verification system [33] requires consistently-tracked faces to capture the diverse pose and spatial-temporal information for analysis. In addition, there is usually more than one person present in the videos, and thus multiple face images from different individuals should be correctly associated across the video frames. Several

recent techniques achieve multiple object tracking by modeling the motion context [34], track management [35], and guided tracking using the confidence map of the detector [36]. Multi-object tracking methods based on tracklet linking [37–39] usually rely on the Hungarian algorithm [40] to optimally assign the detected bounding boxes to existing tracklets. Roth *et al.* [38] adapted the framework of multi-object tracking methods based on tracklet linking approach to track multiple faces; Several face-specific metrics and constraints have been introduced to enhance the reliability of face tracking. A recent study [41] proposed to manage the tracks generated by a continuous face detector without relying on long-term observations. In unconstrained scenarios, the camera can be affected by abrupt movements, which makes consistent tracking challenging. Du *et al.* proposed a conditional random field (CRF) framework for face association in two consecutive frames by utilizing the affinity of facial features, location, motion, and clothing appearance [42]. Our face association method utilizes the KLT tracker to track the face initiated from the face detection. We continuously update the face tracking for every fifth frame using the detected faces. The tracklet linking [39] is utilized to link the fragmented tracklet. We present a robust face association method based on the existing works of [39, 43, 44]. In addition, recently developed object trackers [45–47] and face trackers [48, 49] can be integrated to potentially improve the robustness of face association method. More details are presented in Section 5.2.1.3

2.2 Still/Video Face Recognition: identification and verification

General speaking, there are two major components for a face identification/verification system: (1) robust feature representation and (2) classification model/similarity measure. Due to significant amount of related works in the literature, we briefly review several recent relevant works on face identification and verification as follows.

2.2.1 Robust Feature Representation

Learning invariant and discriminative feature representation is the first step for a face identification/verification system. It can be broadly divided into two categories: (1) hand-crafted features, and (2) feature representation learned from data.

2.2.1.1 Hand-Crafted Feature

Ahonen *et al.* [50] showed that Local Binary Pattern (LBP) is effective for face recognition. Several variants of LBP such as Local Ternary Patterns (LTP) [51] and three-patch LBP (TP-LBP) [52] have been proposed. Gabor wavelets [53] [54] have also been widely used to encode multi-scale and multi-orientation information for face images. Chen *et al.* [55] demonstrated good results for face verification using the high-dimensional multi-scale LBP features extracted from patches around facial landmarks. Ding *et al.* [56] proposed a new texture descriptor called Dual Cross Patterns (DCP) and extracted multi-scale DCP from patches around facial landmarks to compose a high-dimensional feature representation for face recognition.

2.2.1.2 Feature Representation learned from data

Simonyan *et al.* [57] and Parkhi *et al.* [58] applied the Fisher vector (FV) encoding to generate over-complete and high-dimensional feature representation for still and video-based face recognition. Lu *et al.* [59] proposed a dictionary learning framework in which the sparse codes of local patches generated from local patch dictionaries are pooled to generate a high-dimensional feature vector. The high-dimensionality of feature vectors makes these methods hard to train and scale to large datasets. However, advances in deep learning methods have shown that compact and discriminative representation can be learned using DCNNs trained using very large datasets. Taigman *et al.* [60] learned a DCNN model on the frontalized faces generated with a general 3D shape model from a large-scale face dataset and achieved better performance than many traditional face verification methods. In contrast, Sun *et al.* [61] [62] achieved the results surpassing human performance for face verification on the LFW dataset using an ensemble of 25 simple DCNN with fewer layers trained on weakly aligned face images from a much smaller dataset. Schroff *et al.* [11] adapted a state-of-the-art deep architecture in object recognition to face recognition and trained on a large-scale unaligned private face dataset with the triplet loss. Parkhi *et al.* [63] trained a very deep convolutional network based on VGGNet for face verification and demonstrated impressive results.This method also achieved top performances on face verification problems. These works essentially demonstrate the effectiveness of the DCNN model for feature learning and detection/recognition/verification problems.

2.2.2 Classification Model

The classification model for most video-based face recognition algorithms can be classified into two categories: (1) frame-based and (2) image set-based. In addition, similarity measure learning is applicable for both still and video face recognition. We briefly summarize related works as follows.

2.2.2.1 Frame-based Approach

For this category, besides features (*e.g.*, SIFT, LBP) derived from the image intensity data, the temporal (*e.g.*, motion) and spatial-temporal information between cropped faces in a video is usually utilized and encoded in the model to perform recognition tasks. For example, Zhou *et al.* [64] proposed a tracking-and-recognition approach which lowers the uncertainties of tracking and recognition simultaneously in a unified probabilistic framework. Lee *et al.* [65] learned the nonlinear appearance manifold from face videos to handle both tracking and recognition in a unified framework. In addition, a Hidden Markov Model [66] has been also proposed to make use of the temporal information. However, the performance of these approaches is greatly affected by tracking accuracy. Poor tracking will introduce background noise into the model and adversely affect the recognition rates.

2.2.2.2 Image Set-based Approach

In this approach, each face video is transformed into an unordered set of images which implies no temporal information is used. The set of images for a subject is usually

represented using a subspace model. Then, recognition is done by measuring the distance between subspaces. Turaga *et al.* [67] presented a statistical method for video-based face recognition which constructed the face subspaces by performing standard PCA for face videos and using tools from Riemannian geometry of the Grassmann manifold to measure the distance between two faces. Cevikalp *et al.* [68] modeled face image sets using affine or convex hull, and Wang *et. al.* [69] modeled them using covariance matrix to encode the underlying manifold structure. Hu *et al.* [70] improved the affine subspace model by enforcing the sparsity constraint and used it to measure between-set dissimilarity which is the distance between sparse approximated nearest points of two image sets. Recently, Chen *et al.* [71] used K-SVD [72] to learn a compact and representative dictionary for each video and made use of the reconstruction errors of test videos using the learned video dictionaries for face identification and verification tasks. The approach is simple and efficient, especially suitable for large-scale video-based face recognition.

2.2.2.3 Metric Learning

The similarity measure is the other key component in a face verification system. Due to the large number of metric learning approaches in the literature, we briefly review several works on learning a discriminative metric for verification problems. Guillaumin *et al.* [73] proposed to learn two robust distance measures: Logistic Discriminant-based Metric Learning (LDML) and Marginalized kNN (MkNN). The LDML method learns a distance by performing a logistic discriminant analysis on a set of labeled image pairs and the MkNN method marginalizes a k-nearest-neighbor classifier to both images of

the given test pair using a set of labeled training images. Weinberger *et al.* [74] proposed Large Margin Nearest Neighbor(LMNN) metric which enforces the large margin constraint among all triplets of labeled training data. Taigman *et al.* [75] learned the Mahalanobis distance for face verification using the Information Theoretic Metric Learning (ITML) method proposed in [76]. Wolf *et al.* [77] proposed the one-shot similarity (OSS) kernel based on a set of pre-selected reference images mutually exclusive to the pair of images being compared and training a discriminative classifier between the test image and the new reference set. Kumar *et al.* [78] proposed two classifiers for face verification: attribute classifier and simile classifiers. Attribute classifiers are a set of binary classifiers used to detect the presence of certain visual concepts where visual concepts are defined in advance. Simile classifiers were trained to measure the similarities of facial parts of a person to specific reference people. Chen *et al.* [79] proposed a joint Bayesian approach for face verification which models the joint distribution of a pair of face images instead of the difference between them, and the ratio of between-class and within-class probabilities is used as the similarity measure. Hu *et al.* [80] learned a discriminative metric within the deep neural network framework. Huang *et al.* [81] learned a projection metric over a set of labeled images which preserves the underlying manifold structure. Schroff *et al.* [11] and Parkhi *et al.* [63] optimized the DCNN parameters based on the triplet loss which directly embeds the DCNN features into a discriminative subspace and presented promising results for face verification.

2.3 Face Related Application: Facial Age Estimation

For the DCNN model, Donahue *et al.* [12] and Yosinski *et al.* [82] demonstrated that the pre-trained DCNN model can be generalized to other vision tasks by fine-tuning it on the new task. In this dissertation, we focus on finetuning the pre-trained DCNN model for face recognition to the facial age estimation task. We briefly review the related works below.

Most of the age estimation methods proposed earlier have focused on using shape or textural features. These features are then fed to a regression method or a classifier to estimate the apparent age [83–86].

Holistic approaches usually adopt subspace-based methods, while feature-based approaches typically extract different facial regions and compute anthropometric distances. Geometry-based methods [84, 85] are inspired by studies in neuroscience, which suggest that facial geometry strongly influences age perception [85]. As such, these methods address the age estimation problem by capturing the face geometry, which refers to the location of 2D facial landmarks on images. Recently, Wu *et al.* [86] proposed an age estimation method that presents the facial geometry as points on a Grassmann manifold. To solve the regression problem on the Grassmann manifold, [86] then used the differential geometry of the manifold. However, the Grassmannian manifold-based geometry method suffers from a number of drawbacks. First, it heavily relies on the accuracy of landmark detection step, which might be difficult to obtain in practice. For instance, if an image is taken from a bearded person, then detecting landmarks would become a very challenging task. In addition, different ethnic-groups usually have slightly different face geometry,

and to appropriately learn the age model, a large number of samples from different ethnic groups is required.

Unlike the traditional methods discussed, the proposed method is based on DCNN to encode the age information from a given image. Recent advances in deep learning methods have shown that compact and discriminative image representations can be learned using DCNN from very large datasets [87]. There are various neural-network-based methods, which have been developed for facial age estimation [88–90] . However, as the number of samples for estimating the apparent age task is limited, (i.e. not enough to properly learn discriminative features, unless a large number of external data is added), the traditional neural network methods often fail to learn an appropriate model.

Thukral *et. al.* [91] proposed a cascaded approach for apparent age estimation based on classifiers using the naive-Bayes approach and a support vector machine (SVM) and regressors using the relevance vector machine (RVM). However, the difference between [91] and the proposed approach is that we leverage the rich information contained in the DCNN model pre-trained using a large-scale face dataset for age estimation. Also, the proposed error correction module mitigates the influences of the errors made at initial classification stage.

Chapter 3: Dictionary-based Video Face Recognition Using Dense Multi-scale Facial Landmark Features

3.1 Overview

Motivated by the successes of high-dimensional facial features in still-face recognition [55], sparse representation [92] and dictionary learning for video-based face recognition [93] [71] [94], we propose a dictionary-based approach using dense high-dimensional feature for unconstrained video-to-video face identification problems. We first segment the face videos into K partitions and extract multi-scale features from patches centered at detected dense facial landmarks. Then, we learn a compact and representative dictionary from dense features for each partition and form a video dictionary for each video by concatenating sub-dictionaries. Finally, the learned video dictionaries are used for face identification. Moreover, because the dictionary for each training video is learned independently during the training phase, our approach can thus be easily parallelized in training and testing stages. This makes our approach attractive for addressing the large-scale video-based face recognition problems. Fig. 3.1 gives an overview of our method.

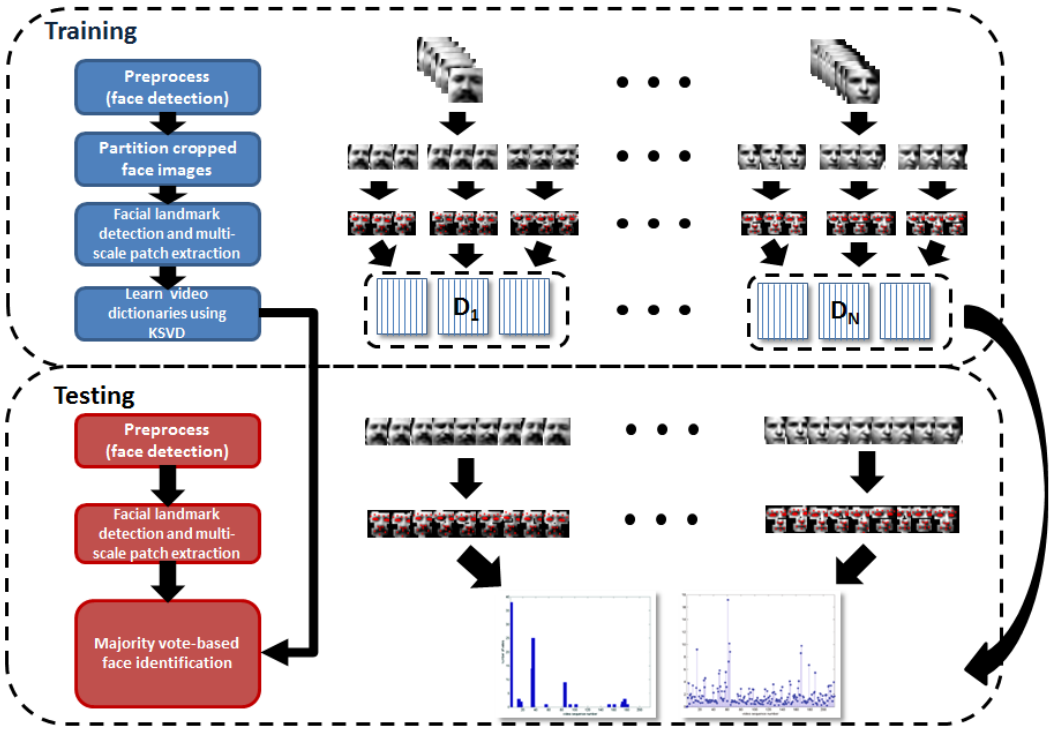


Figure 3.1: An overview for our video-based face identification system.

3.2 Proposed Approach

In this section, we describe the construction of a video dictionary using high-dimensional dense facial landmark features and its application to face identification problems.

3.2.1 Constructing Video Dictionary Using Dense Multi-scale Facial Landmark Features

The training phase of our method consists of three main stages: video partitioning, multi-scale landmark feature extraction and video dictionary learning. In what follows, we describe them in detail.

Video partitioning: Due to the high variability of faces within a video and face tracking accuracy, we find that segmenting a video into different partitions usually improves recognition accuracy. A K-means clustering type of algorithm is used to segment the videos [71] [95] which incrementally adds each cropped face into a partition with the minimum ratio of within-partition similarity over between-partition similarity.

Dense landmarks and multiple-scale features: It was shown in [55] that multi-scale features centered around facial landmarks contain strong discriminative information and the recognition performance improves as the dimensionality of the feature vector is increased. We extract multi-scale patches centered at facial landmarks of inner faces (*i.e.*, landmarks at eye brows, eyes, nose, and mouth corners. 26 landmarks in total are used in our work) and concatenate them together to form a high-dimensional feature vector. With recent progress in face alignment, there are numerous approaches providing accurate and dense facial landmark detection [96] [97]. We adopt [29] because of its excellent performance on low-resolution and lower-quality face images[1]. Detected landmarks and extracted features are shown in Fig.3.2. However, unlike still-face recognition, directly applying the approach in [55] to video-to-video face recognition is infeasible because the concatenation of feature vectors extracted from each frames in a video yields extremely high-dimensional feature vector (*i.e.*, imagine a video with 100 frames can result in a 100 times long feature vector). A compact and representative model has to be learned to remove noisy and irrelevant features.

[1] https://sites.google.com/site/akshayasthana/clm-wild-code.

Figure 3.2: For illustration purposes, we visualize the single-scale patch image for the MBGC dataset by assembling all 5×5-pixel patches centered at 26 facial landmarks points together.

Video dictionary: Various algorithms have been proposed in the literature for learning compact and representative dictionaries. One of the well-known algorithm is the K-SVD algorithm [72]. For each partition, we apply the K-SVD algorithm to construct a dictionary which not only captures variations caused by changes in pose and illumination but also reduces temporal redundancy. Let $\mathbf{D}^i_{j,k}$ be the dictionary and $\mathbf{G}^i_{j,k} = [\mathbf{g}^i_{j,k,1}\ \mathbf{g}^i_{j,k,2}\ \ldots]$ be the feature matrix for the kth partition of the jth face video for the ith subject where each column $\mathbf{g}^i_{j,k,l}$ is the extracted dense multi-scale feature for lth face in the kth partition of the jth video. In the K-SVD formulation, the dictionary and sparse coefficients are learned through iteratively minimizing the following reconstruction errors by fixing $\mathbf{D}^i_{j,k}$ and $\mathbf{X}^i_{j,k}$ in turn.

$$(\hat{\mathbf{D}}^i_{j,k}, \hat{\mathbf{X}}^i_{j,k}) = \underset{\mathbf{D}^i_{j,k}, \mathbf{X}^i_{j,k}}{\operatorname{argmin}} ||\mathbf{G}^i_{j,k} - \mathbf{D}^i_{j,k}\mathbf{X}^i_{j,k}||^2_F \ s.t.\ \forall l, ||\mathbf{x}_l||_0 \leq T_0, \tag{3.1}$$

where $T_0 \in \mathbb{N}$ is the sparsity constraint and $\mathbf{x}_l$ is the lth column of sparse coefficient matrix $\mathbf{X}^i_{j,k}$. $||\cdot||_0$ is the zero-norm which counts the number of nonzero entries, and $||\cdot||_F$ is the Frobenius norm. Finally, the video dictionary $\mathbf{D}^i_j$ for the jth video of ith subject can be obtained via concatenating all sub-dictionaries learned from the corresponding K partitions

$$\mathbf{D}^i_j = [\mathbf{D}^i_{j,1}\ \mathbf{D}^i_{j,2}\ \ldots\ \mathbf{D}^i_{j,K}]. \tag{3.2}$$

After the video dictionaries are learned, in the testing phase we first do the same image preprocessing as in training and extract the multi-scale features for each cropped face image. Then, we perform face identification as discussed in the following subsections.

3.2.2 Face Identification

Let $\mathbf{P}$ represent the set of the entire gallery videos (*i.e.*, training videos) and $\mathbf{Q}$ represent the set of the entire query videos (*i.e.*, test videos) where $\mathbf{Q}^m$ is the mth query video with $m = 1, 2, \ldots, |\mathbf{Q}|$. In addition, the feature vector for lth frame in mth query video is denoted as q^m_l where $l = 1, 2, \ldots, |\mathbf{Q}^m|$. The learned dictionary for the pth gallery videos is denoted as $\mathbf{D}_p$ where $p = 1, 2, \ldots, |\mathbf{P}|$. The original identification problem can be converted as finding the gallery video dictionary which produces the minimum reconstruction error for $\mathbf{q}^m_l$:

$$\hat{p} = \underset{p}{\operatorname{argmin}}\, ||\mathbf{q}^m_l - \mathbf{D}_p\mathbf{D}^\dagger_p\mathbf{q}^m_l||_2, \tag{3.3}$$

where $\mathbf{D}^\dagger_p = (\mathbf{D}^T_p\mathbf{D}_p)^{-1}\mathbf{D}^T_p$ is the pseudo inverse of $\mathbf{D}_p$ and $\mathbf{D}_p\mathbf{D}^\dagger_p\mathbf{q}^m_l$ is the projection of $\mathbf{q}^m_l$ onto the subspace spanned by the atoms of $\mathbf{D}_p$.

Then, the final decision is made for $\mathbf{Q}^m$ through aggregating the voting results from its frames as

$$p^* = \underset{p}{\operatorname{argmax}} \ \mathbf{C}_p, \tag{3.4}$$

where $\mathbf{C}_p$ is the total number of the frames in $\mathbf{Q}^m$ voting to the pth gallery video. The subject identity can be decided through the video-to-subject mapping as $i = m(p^*)$.

3.3 Kernel Dictionary-based Video Face Recognition

The faces for each subject usually distribute on a smooth manifold. Nevertheless, in unconstrained settings, factors such as large pose and illumination changes and occlusion often make the situations much more complicated than usual, and the faces of all subjects may thus be not linearly separable to correctly determine the associated subject identities in the original space. For this reason, we extend our framework through kernelizing our dictionary model as in [98] to handle the nonlinearity problem.

Let $\mathbf{\Phi} : \mathbb{R}^d \rightarrow \mathcal{H}$ be a nonlinear mapping from d-dimensional space into a higher-dimensional feature space $\mathcal{H}$. In this chapter, we use the dictionary model $\mathbf{D} = \mathbf{BA}$, where $\mathbf{B}$ is the predefined base dictionary which can be selected to include prior knowledge of data and $\mathbf{A}$ is the atom representation dictionary which can be modified.

For nonlinear case, let $\mathbf{B} = \mathbf{\Phi}(\mathbf{G}^i_{j,k})$ since the dictionary lies in the subspace spanned by the transformed data samples $\mathbf{\Phi}(\mathbf{G}^i_{j,k}) = [\mathbf{\Phi}(\mathbf{g}_{j,k,1}), \ldots, \mathbf{\Phi}(\mathbf{g}_{j,k,M})]$ in $\mathcal{H}$ where $M = |\mathbf{G}^i_{j,k}|$. Then, the dictionary $\mathbf{D}^i_{j,k}$ for the kth partition of jth video of ith subject can be represented as

$$\mathbf{D}^i_{j,k} = \mathbf{\Phi}(\mathbf{G}^i_{j,k})\mathbf{A}^i_{j,k}, \tag{3.5}$$

Then, substitute $\mathbf{\Phi}(\mathbf{G}^i_{j,k})$ for $\mathbf{G}^i_{j,k}$ and (3.5) for $\mathbf{D}^i_{j,k}$ in (3.1). The nonlinear dictionary can thus be learned in the feature space $\mathcal{H}$ via solving the following optimization problem.

$$\begin{aligned}(\hat{\mathbf{A}}^i_{j,k}, \hat{\mathbf{X}}^i_{j,k}) = \underset{\mathbf{A}^i_{j,k}, \mathbf{X}^i_{j,k}}{\operatorname{argmin}} ||\mathbf{\Phi}(\mathbf{G}^i_{j,k}) - \mathbf{\Phi}(\mathbf{G}^i_{j,k})\mathbf{A}^i_{j,k}\mathbf{X}^i_{j,k}||^2_F, \\ s.t.\ \forall l\ ||\mathbf{x}_l||_0\ \leq\ T_0,\end{aligned} \tag{3.6}$$

Furthermore, since $||\mathbf{U}||^2_F = \mathbf{tr}(\mathbf{U}^T\mathbf{U})$, the objective function in (3.6) can be rewritten as

$$\begin{aligned}&||\mathbf{\Phi}(\mathbf{G}^i_{j,k}) - \mathbf{\Phi}(\mathbf{G}^i_{j,k})\mathbf{A}^i_{j,k}\mathbf{X}^i_{j,k}||^2_F \\ &= ||\mathbf{\Phi}(\mathbf{G}^i_{j,k})(\mathbf{I} - \mathbf{A}^i_{j,k}\mathbf{X}^i_{j,k})||^2_F \\ &= \mathbf{tr}((\mathbf{I} - \mathbf{A}^i_{j,k}\mathbf{X}^i_{j,k})^T\mathcal{K}(\mathbf{G}^i_{j,k}, \mathbf{G}^i_{j,k})(\mathbf{I} - \mathbf{A}^i_{j,k}\mathbf{X}^i_{j,k})),\end{aligned} \tag{3.7}$$

where $\mathcal{K}(\mathbf{G}^i_{j,k}, \mathbf{G}^i_{j,k})$ is the kernel matrix whose (r, s)th entries can be computed by

$$\mathcal{K}(\mathbf{g}^i_{j,k,r}, \mathbf{g}^i_{j,k,s}) = \mathbf{\Phi}(\mathbf{g}^i_{j,k,r})^T\mathbf{\Phi}(\mathbf{g}^i_{j,k,s}).$$

From this formulation, we observe two points: (1) The kernel matrix $\mathcal{K} \in \mathbb{R}^{M\times M}$ is of finite dimension which ensures the computation is feasible, and (2) $\mathcal{K}$ is the Gram matrix of $\mathbf{\Phi}(\mathbf{G}^i_{j,k})$, so we can simply use Mercer's kernels for $\mathcal{K}$ without explicitly knowing the exact form of the mapping function $\mathbf{\Phi}$. This technique is also referred as kernel trick which is widely used in machine learning to extend the recognition algorithms to handle data nonlinearity. Commonly used Mercer's kernels include (1) the polynomial kernel

$$\mathcal{K}(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + c)^d$$

where c and d are the bias and degree for polynomial kernel respectively (*i.e.*, linear kernel is the special case for polynomial kernel where the bias term is equal to 0 and the degree is equal to 1.), and (2) the Gaussian kernel

$$\mathcal{K}(\mathbf{x}, \mathbf{y}) = exp(-\frac{||\mathbf{x} - \mathbf{y}||^2}{\sigma}),$$

where σ is the variance for Gaussian kernels.

Likewise, we obtain the nonlinear video dictionary for each subject video through concatenating learned sub-dictionaries from each partition.

$$\begin{aligned} \mathbf{D}_j^i &= [\mathbf{D}_{j,1}^i, \ldots, \mathbf{D}_{j,K}^i] \\ &= [\mathbf{\Phi}(\mathbf{G}_{j,1}^i)\mathbf{A}_{j,1}^i, \ldots, \mathbf{\Phi}(\mathbf{G}_{j,K}^i)\mathbf{A}_{j,K}^i] \\ &= [\mathbf{\Phi}(\mathbf{G}_{j,1}^i), \ldots, \mathbf{\Phi}(\mathbf{G}_{j,K}^i)] \begin{bmatrix} \mathbf{A}_{j,1}^i & & 0 \\ & \ddots & \\ 0 & & \mathbf{A}_{j,K}^i \end{bmatrix} \\ &= \mathbf{\Phi}(\mathbf{G}_j^i)\mathbf{A}_j^i \end{aligned} \tag{3.8}$$

where $\mathbf{\Phi}(\mathbf{G}_j^i)$ and $\mathbf{A}_j^i$ are the transformed feature and coefficient matrices for the jth video of ith subject, and K is the number of partitions.

3.3.1 Nonlinear Face Identification

Assuming we have P gallery videos in total, we learn a nonlinear dictionary $\mathbf{D}_p = \mathbf{\Phi}(\mathbf{G}_p)\mathbf{A}_p$, for each video where $p = 1, \ldots, P$, and we denote $\mathbf{D}_p$ for some $\mathbf{D}_j^i$ of the jth video of ith subject in the previous section for simplicity. To find the coefficient vector of lth frame of mth query video, $\mathbf{x}_l^m$, which has at most T_0 non-zero entries and minimizes the reconstruction error between $\mathbf{\Phi}(\mathbf{q}_l^m)$ and $\mathbf{\Phi}(\mathbf{G_p})\mathbf{A}_p\mathbf{x}_l^m$, we solve the following optimization problem:

$$\min_{\mathbf{x}_l^m} ||\mathbf{\Phi}(\mathbf{q}_l^m) - \mathbf{\Phi}(\mathbf{G}_p)\mathbf{A}_p\mathbf{x}_l^m||_2^2 \; s.t. \; ||\mathbf{x}_l^m||_0 \leq T_0. \tag{3.9}$$

The solution can be efficiently computed by the Kernel Orthogonal Matching Pursuit (KOMP) approach and the details can be found in [98]. Similarly as the linear case (3.3), we can decide the label, $\hat{p}$, of the frame as the one whose corresponding nonlinear dictionaries produce the minimum reconstruction error.

$$\begin{aligned} \hat{p} &= \text{argmin}_p \, ||\mathbf{\Phi}(\mathbf{q}_l^m) - \mathbf{\Phi}(\mathbf{G}_p)\mathbf{A}_p\mathbf{x}_l^m||_2^2 \\ &= \text{argmin}_p \, \mathcal{K}(\mathbf{q}_l^m, \mathbf{q}_l^m) - 2\mathcal{K}(\mathbf{q}_l^m, \mathbf{G}_p)\mathbf{A}_p\mathbf{x}_l^m + (\mathbf{x}_l^m)^T\mathbf{A}_p^T\mathcal{K}(\mathbf{G}_p, \mathbf{G}_p)\mathbf{A}_p\mathbf{x}_l^m \end{aligned} \tag{3.10}$$

where

$$\mathcal{K}(\mathbf{q}_l^m, \mathbf{G}_p) = \left[\mathcal{K}(\mathbf{q}_l^m, \mathbf{g}_{p,1}), \mathcal{K}(\mathbf{q}_l^m, \mathbf{g}_{p,2}), \ldots, \mathcal{K}(\mathbf{q}_l^m, \mathbf{g}_{p,|\mathbf{G}_p|})\right].$$

To decide the subject label for a query video, we first aggregate the label decisions of each frame in $\hat{\mathbf{C}}_p$ the same as in (3.4). Finally, the label can be attained through the

video to subject mapping $i = m(p^*)$ where $\hat{\mathbf{C}}_{p^*}$ attains the maximum number of frame votes.

3.4 Experimental Results

To evaluate our approach, we present face identification results on the standard Honda/UCSD video dataset [65] and another two well-known public datasets for unconstrained video-based face recognition: (1) Multiple Biometric Grand Challenge (MBGC) [99], and (2) Face and Ocular Challenge Series (FOCS) [100]. We perform our experiments following the experimental design described in [71] [101].

3.4.1 Implementation Details

We used the face detector in OpenCV [14] and IVT [102] for face detection and face tracking respectively to crop the faces from each video. All cropped faces are down-sampled and normalized to 20×20 pixels, and two patch sizes are used for multi-scale feature extraction: (1) 5×5 and (2) 7×7 pixels. In addition, we segment $K = 3$ partitions for each video in the MBGC dataset and the FOCS dataset in all of our experiments. Prior to dictionary learning, we augment the feature matrix for each partition by adding more multi-scale patch features which are extracted via shifting the original bounding boxes of patches by one or two pixels to all directions or rotating them with a small angle. This helps the partition step in assigning video frames to learn an improved dictionary and helps in reducing the noise caused by tracking and landmark detection. The same augmentation is also applied to query videos before recognition. For our kernel-based

Figure 3.3: The upper row shows the example frames from the MBGC walking sequences in four different scenarios. Similarly, the bottom row presents the example frames from the FOCS UT-Dallas walking videos.

approach, we use the polynomial kernel by setting the degree to 2 and bias to 0 for all our experiments in this work.

3.4.2 Multiple Biometric Grand Challenge

In the MBGC video version 1 dataset (Notre Dame dataset), there are 146 subjects in total, and videos for each subject are available in two formats: standard definition (SD, 720 $\times$ 480 pixels) and high definition (HD, 1440 $\times$ 1080 pixels). It consists of 399 walking sequences where 201 of them are in SD format and 198 in HD, and 371 activity sequences where 185 in SD and 186 in HD. For the walking sequences as illustrated in Fig. 3.3, subjects usually walk toward the camera and keep their faces frontal with respective to it for most of the time and turn their face to the left or right at the end. On the contrary, the activity sequences contains most non-frontal views for each subject. The challenge for the dataset includes blurred faces caused by motion, frontal and non-frontal faces in shadow which also induce face tracking difficulty to crop faces from the video.

We conduct leave-one-out identification experiments on three subsets of the cropped face images acquired from walking videos and present the identification accuracy in Table 3.1. Our proposed method outperforms other approaches. The three subsets are S2,

S3, and S4, respectively where S2 is the set of subjects who have at least two face videos available, S3 at least three available, and S4 at least four available (S2: 144 subjects, 397 videos in total, S3: 55 subjects, 219 videos in total, and S4: 54 subjects, 216 videos).

MBGC walking videos	WGCP [67]	SANP [70]	DFRV [71]	KSRV [101]	Ours	Ours with kernel
S2	63.79	83.88	85.64	86.65	89.17	**99.24**
S3	74.88	84.02	88.13	88.58	89.04	**99.08**
S4	75	84.26	88.43	88.89	89.35	**99.07**
Average	71.22	84.05	87.40	88.04	89.19	**99.13**

Table 3.1: Identification rate for leave-one-out face identification experiments for the MBGC walking videos. Our method achieves the best results.

From the table, the proposed approach achieves better results than DFRV which essentially demonstrates the effectiveness of dense multi-scale facial landmark features.

3.4.3 Face and Ocular Challenge Series

The FOCS UT-Dallas dataset contains 510 walking (frontal-face) and 506 activity (non-frontal face) video sequences for 295 subjects in the resolution, 720 $\times$ 480 pixels. The sequences were acquired on different days. For the walking sequences, subjects stand far away from the camera originally, and then walk toward the camera keeping their face in frontal pose and turn away at the end. For the dataset, we conducted the same leave-one-out tests on 3 subsets: S2 (189 subjects, 404 videos), S3 (19 subjects, 64 videos), and S4 (6 subjects, 25 videos) for UT-Dallas walking videos.

The results are shown in Table 3.2. Our approach performs the best when compared to other competitive methods.

UT-Dalas walking videos	PM [103] [67]	WGCP [67]	SANP [70]	DFRV [71]	Ours	Ours with kernel
S2	38.12	53.22	48.27	59.90	61.39	**68.81**
S3	60.94	70.31	60.94	78.13	79.69	**85.94**
S4	64	76	68.00	80.00	84.00	**88.00**
Average	54.35	66.51	59.07	72.68	75.02	**80.92**

Table 3.2: Identification rate for leave-one-out face identification experiments for the FOCS UT-Dallas walking videos. Our method achieves the best results.

3.4.4 Honda/UCSD Dataset

Honda Set length	MMA [104]	AHISD [68]	CHISD [68]	SANP [70]	DFRV [71]	Ours	Ours with kernel
50 frames	74.36	87.18	82.05	84.62	89.74	87.18	**97.44**
100 frames	94.87	84.62	84.62	92.31	97.44	97.44	**100**
full length	97.44	89.74	92.31	100	97.44	97.44	**100**
Average	88.89	87.18	86.33	92.31	94.87	94.02	**99.15**

Table 3.3: Identification rate for the Honda videos. Our dense feature representation with kernel dictionary achieves the best results.

The third experiments is conducted on the Honda/UCSD dataset. The dataset is the standard benchmark used in various image-set based face recognition works. There are 59 videos for 20 subjects for the dataset. We follow the same setting used in [70] which contains three cases based on the available maximum number of cropped faces per video: (1) 50 frames, (2) 100 frames, and (3) all available frames. The results are presented in Table 3.3. Our approach with the linear kernel works comparable with the approach, and the kernelized one achieves the best results. One possible reason is due to the small size of this dataset.

3.5 Summary

In this chapter, we demonstrated that the proposed dictionary approach with dense facial landmark features is effective for unconstrained video-based face identification. Experiments using the Honda/UCSD, MBGC, and FOCS datasets show that high-dimensional features extracted from multi-scale patches centered around the detected dense facial landmarks provide strong discriminative information upon different pose and illumination conditions among subjects, and video dictionaries provide an efficient and feasible way to utilize the high-dimensional features for large-scale unconstrained video-based face recognition.

Chapter 4: Landmark-based Fisher Vector Representation for Video-based Face Verification

4.1 Overview

To handle large variations in pose, expression and illumination, extracting invariant and discriminative representation from face images/videos is an important issue. Chen *et al.* [55] have shown that the high-dimensional multi-scale Local Binary Pattern (LBP) descriptors extracted from local patches centered at each facial landmarks have strong discriminative power for the still-face recognition problem. However, directly applying this idea to videos is infeasible because of the high dimensionality of the feature representation. On the other hand, the Fisher Vector (FV) representation is one of many bag-of-visual-word encoding methods, originally proposed for object recognition problem and subsequently shown to work well for face verification problems [57] [58]. Even though FV descriptors are compact for videos, their dimension is still high and increases linearly with the number of components in the Gaussian Mixture Model (GMM). More components in GMM representation usually allow FVs to encode more discriminative information from image and video data. However, having many mixture components may be impractical for large face databases. Motivated by the successes of these two

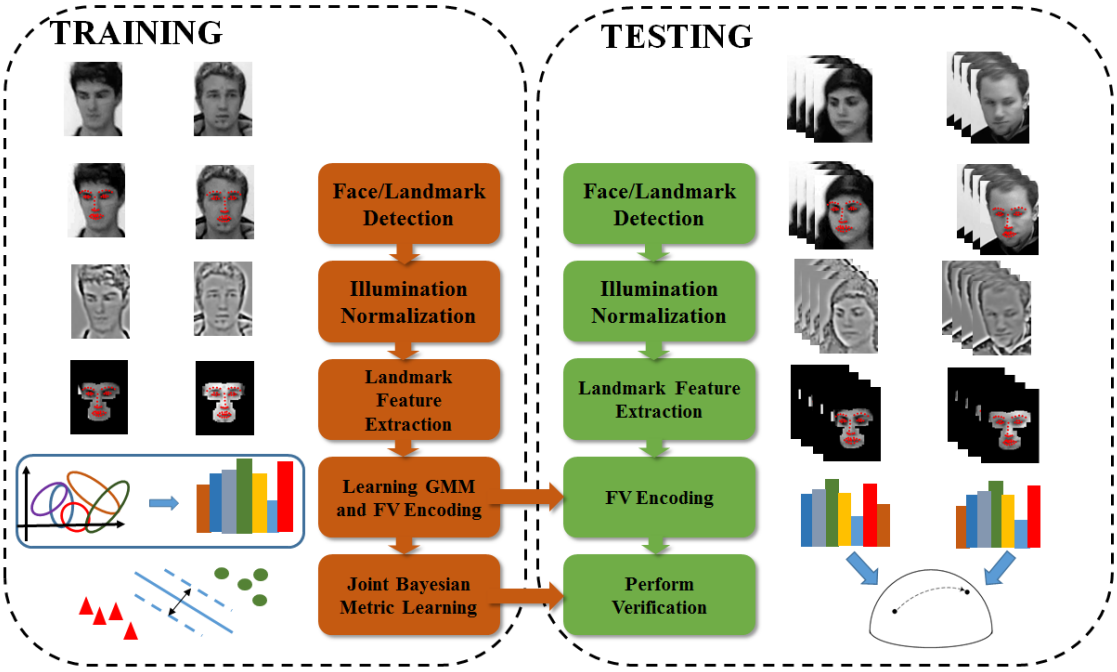


Figure 4.1: An overview for our landmark-based Fisher vector video-based face verification algorithm.

approaches, we propose a landmark-based FV representation for video-based face verification. Instead of learning the mixture model from the dense features of the whole face, we fit a Gaussian model for each landmark with multi-scale dense features extracted from patches centered at each landmark. In this way, we can greatly reduce the number of mixture components and the dimensionality of the FVs while preserving sufficient discriminative power.

4.2 PROPOSED APPROACH

Our method can be divided into two stages: training and testing stages. For training, we use the well-known "Label Face in the Wild" (LFW) dataset [105]. First, we apply preprocessing steps to detect faces, facial landmarks and to normalize the face im-

ages/videos. Then, we extract multi-scale dense SIFT features around each landmark and learn a Gaussian model for each landmark using the mean and diagonal sample covariance of the features. After feature extraction, we perform the FV encoding and train a similarity measure using the augmented face pairs (*i.e.* we generate positive and negative pairs using the identity information available in the unrestricted setting of LFW). For testing, we use the learned metric on our proposed feature representation to compute the similarity of each test pair of the face images/videos. Fig. 4.1 presents an overview of our method. In the following subsections, we describe in detail each step used in training and testing stages.

4.2.1 Preprocessing

Before performing feature extraction and metric learning steps, we apply the following preprocessing steps to normalize the face data:

Landmark detection: We perform landmark detection for face alignment and for landmark-based feature representation. Approaches proposed in [29] and subsequent work [106] are adopted because of their computational efficiency and excellent performance on low-resolution and lower-quality face images/videos. We use the detected landmarks to align each face into the canonical coordinates using similarity transform. After alignment, the face image resolution is 63×80 pixels, and the distance between centers of two eyes is about 10 pixels.

Figure 4.2: The first row shows the original image before preprocessing. The second row is the image after illumination normalization. The final row demonstrates the facial landmarks and patches used in this chapter.

Illumination normalization: Local block-wise illumination normalization approaches, such as self-quotient image (SQI) [107] which divides each pixel value by the weighted average of its neighborhood, have shown better illumination normalization performance for face recognition than histogram equalization which enhances the dynamic range by adjusting the intensity distribution of the entire image. Therefore, we adopt the SQI approach proposed by Tan *et al.* [51] which takes the Gamma correction, difference of Gaussian filtering, masking, and contrast equalization into consideration for image normalization. The normalized results are presented in Fig. 4.2.

4.2.2 Landmark-based Fisher vector face representation

In this subsection, we show how to extract the proposed landmark-based FV face representation (LFRV) and to apply metric learning on the extracted representation to compute the face similarity of a pair of face images/videos.

Fisher vector encoding: Fisher vector is one of bag-of-visual-word encoding methods which aggregates a large set of local features into a high-dimensional vector. In general, the FV is extracted by fitting a parametric generative model for the features and encoding them using the derivatives of the log-likelihood of the learned model with respect to the model parameters. As in [108], a Gaussian mixture model (GMM) with diagonal covariances is used here. In addition, the first-and second-order statistics of the features with respect to each component are computed as follows:

$$\mathbf{\Phi}_{ik}^{(1)} = \frac{1}{N\sqrt{w_k}} \sum_{p=1}^{N} \alpha_k(\mathbf{v}_p) \left(\frac{\mathbf{v}_{ip} - \boldsymbol{\mu}_{ik}}{\boldsymbol{\sigma}_{ik}} \right) \tag{4.1}$$

$$\mathbf{\Phi}_{ik}^{(2)} = \frac{1}{N\sqrt{2w_k}} \sum_{p=1}^{N} \alpha_k(\mathbf{v}_p) \left(\frac{(\mathbf{v}_{ip} - \boldsymbol{\mu}_{ik})^2}{\boldsymbol{\sigma}_{ik}^2} - 1 \right) \tag{4.2}$$

$$\alpha_k(\mathbf{v}_p) = \frac{w_k \exp[-\frac{1}{2}(\mathbf{v}_p - \boldsymbol{\mu}_k)^T \mathbf{\Sigma}_k^{-1} (\mathbf{v}_p - \boldsymbol{\mu}_k)]}{\sum_i^K w_i \exp[-\frac{1}{2}(\mathbf{v}_p - \boldsymbol{\mu}_i)^T \mathbf{\Sigma}_i^{-1} (\mathbf{v}_p - \boldsymbol{\mu}_i)]}, \tag{4.3}$$

where w_k, $\boldsymbol{\mu}_k$, $\mathbf{\Sigma}_k = diag(\boldsymbol{\sigma}_{1k}, ..., \boldsymbol{\sigma}_{dk})$ are the weights, means, and diagonal covariances of the kth mixture component of the GMM. Here, $\mathbf{v}_p \in \mathbb{R}^{d \times 1}$ is the pth feature vector and N is the number of feature vectors. The parameters can be learned from the training data using the EM algorithm. $\alpha_k(\mathbf{v}_p)$ is the weight of $\mathbf{v}_p$ belonging to the kth mixture component. The final FV, $\mathbf{\Phi}(\mathbf{I})$, of an image $\mathbf{I}$ is obtained by concatenating all the $\mathbf{\Phi}_k^{(1)}$ and $\mathbf{\Phi}_k^{(2)}$s into a high-dimensional vector $\mathbf{\Phi}(\mathbf{I}) = [\mathbf{\Phi}_1^{(1)}, \mathbf{\Phi}_1^{(2)}, ..., \mathbf{\Phi}_K^{(1)}, \mathbf{\Phi}_K^{(2)}]$, whose dimensionality is $2Kd$ where K is the number of mixture components and d is the dimensionality of the extracted features.

In this work, we use the dense SIFT features as local features. To incorporate spatial information, we augment each extracted SIFT feature with the normalized x and y coor-

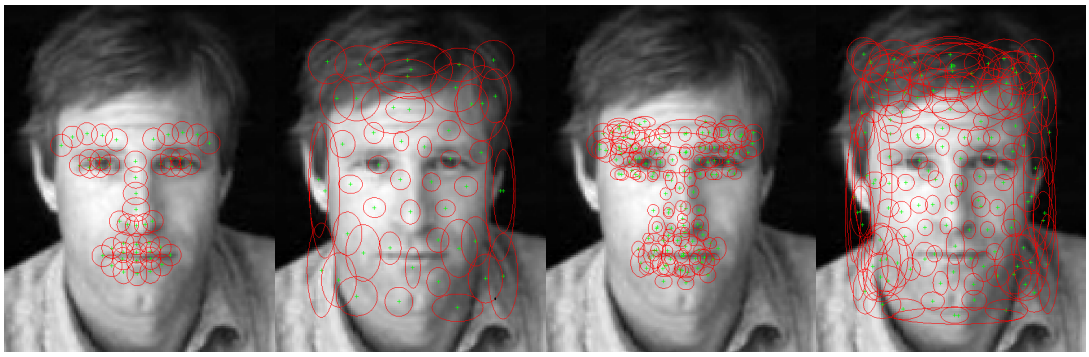
dinates [109] [57] as $[\mathbf{a}_{xy}, \frac{x}{w} - \frac{1}{2}, \frac{y}{h} - \frac{1}{2}]^T$ where $\mathbf{a}_{xy}$ is the SIFT descriptor at (x, y), and w and h are the width and height of the image, respectively. (*i.e.* For K, we use 49 and 128. For d, it is 130 after augmentation.) In addition, FV is further processed with signed square-rooting and L_2 normalization as suggested in [108] for improved performance.

Dense landmark features extraction: We extract dense root-SIFT features at three scales from 16 $\times$ 16-pixel patches centered at each facial landmark of inner faces with a scaling factor of $\sqrt{2}$ (*i.e.*, 49 landmarks are used here). For training, we aggregate the extracted features around each landmark and take the mean and diagonal sample covariance, $\mathbf{\Sigma}_k = diag(\boldsymbol{\sigma}_{1k}, ..., \boldsymbol{\sigma}_{dk})$, to fit a Gaussian for each landmark as follows:

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{p=1}^{N_k} \mathbf{v}_p, \, w_k = \frac{1}{K}, \quad \boldsymbol{\sigma}_{ik} = \frac{1}{N_k - 1} \sum_{p=1}^{N_k} (\mathbf{v}_{ip} - \boldsymbol{\mu}_{ik})^2,$$

where N_k and $\mathbf{v}_p$ are respectively the number of features and SIFT features extracted from the patch centered at kth landmark. The fitted Gaussians are illustrated in Fig. 4.3.

For testing, we aggregate the extracted features with augmented spatial information into a feature matrix, $\mathbf{F} \in \mathbb{R}^{130 \times N_F}$ for each frame, where N_F is the total number of aggregated features. Because some patches overlaps, we take the union of them to remove the duplicate features. Detected landmarks and patches for feature extraction are shown in Fig. 4.2. Then, we perform FV encoding for each frame within a video and average all the FVs into one for each video. (*i.e.* the other choice is to use pooling.)



(a) (b) (c) (d)

Figure 4.3: (a) and (b) illustrate the GMM with 49 components learned from 49 facial landmarks and from the whole image, respectively. (c) and (d) show the GMM with 128 components learned from the neighborhood regions of 49 facial landmarks using EM algorithm and learned from the entire image respectively.

4.2.3 Joint Bayesian Metric Learning

The joint Bayesian method has been shown good performance for face verification task [79] [110]. Instead of modeling the difference vector between two faces, the approach directly models the joint distribution of feature vectors of both ith and jth images, $\{\mathbf{x}_i, \mathbf{x}_j\}$, as a Gaussian. Let $P(\mathbf{x}_i, \mathbf{x}_j|H_I) \sim N(0, \mathbf{\Sigma}_I)$ when $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same class, and $P(\mathbf{x}_i, \mathbf{x}_j|H_E) \sim N(0, \mathbf{\Sigma}_E)$ when they are from different classes. In addition, each face vector can be modeled as, $\mathbf{x} = \boldsymbol{\mu} + \boldsymbol{\epsilon}$, where $\boldsymbol{\mu}$ stands for the identity and $\boldsymbol{\epsilon}$ for pose, illumination, and other variations. Both $\boldsymbol{\mu}$ and $\boldsymbol{\epsilon}$ are assumed to be independent zero-mean Gaussian distributions, $N(0, \mathbf{S}_\mu)$ and $N(0, \mathbf{S}_\epsilon)$, respectively. Then, the covariances for intra-class, $\mathbf{\Sigma}_I$, and for inter-class, $\mathbf{\Sigma}_E$, can be derived as follows

$$\mathbf{\Sigma}_I = \begin{bmatrix} \mathbf{S}_\mu + \mathbf{S}_\epsilon & \mathbf{S}_\mu \\ \mathbf{S}_\mu & \mathbf{S}_\mu + \mathbf{S}_\epsilon \end{bmatrix}, \mathbf{\Sigma}_E = \begin{bmatrix} \mathbf{S}_\mu + \mathbf{S}_\epsilon & 0 \\ 0 & \mathbf{S}_\mu + \mathbf{S}_\epsilon \end{bmatrix}. \tag{4.4}$$

It was shown in [79] that the log likelihood ratio of intra- and inter-classes, $r(\mathbf{x}_i, \mathbf{x}_j)$, which has a closed-form solution can be computed as follows:

$$r(\mathbf{x}_i, \mathbf{x}_j) = log \frac{P(\mathbf{x}_i, \mathbf{x}_j|H_I)}{P(\mathbf{x}_i, \mathbf{x}_j|H_E)} = \mathbf{x}_i^T \mathbf{M} \mathbf{x}_i + \mathbf{x}_j^T \mathbf{M} \mathbf{x}_j - 2\mathbf{x}_i^T \mathbf{R} \mathbf{x}_j \tag{4.5}$$

where

$$\mathbf{M} = (\mathbf{S}_\mu + \mathbf{S}_\epsilon)^{-1} - (\mathbf{F} + \mathbf{R}) \tag{4.6}$$

$$\begin{bmatrix} \mathbf{F} + \mathbf{R} & \mathbf{R} \\ \mathbf{R} & \mathbf{F} + \mathbf{R} \end{bmatrix} = \mathbf{\Sigma}_I^{-1}. \tag{4.7}$$

where $\mathbf{M}$ and $\mathbf{R}$ are negatively semi-definite matrices. The equation can be written as $(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j) - 2\mathbf{x}_i^T (\mathbf{R} - \mathbf{M}) \mathbf{x}_j$. Instead of using the EM algorithm to estimate $\mathbf{S}_\mu$ and $\mathbf{S}_\epsilon$, we optimize the closed-form distance in a large-margin framework with hinge loss. However, directly learning $\mathbf{M} \in \mathbb{R}^{D \times D}$ and $\mathbf{R} \in \mathbb{R}^{D \times D}$ are intractable because of the high dimensionality of FVs where $D = 2Kd$. Thus, we let $\mathbf{M} = \mathbf{H}^T \mathbf{H}$ and $\mathbf{B} = (\mathbf{R} - \mathbf{M}) = \mathbf{V}^T \mathbf{V}$ where $\mathbf{H} \in \mathbb{R}^{r \times D}$ and $\mathbf{V} \in \mathbb{R}^{r \times D}$ and choose $r = 128 \ll D$ in our work. We solve the following optimization problem

$$\underset{\mathbf{H}, \mathbf{V}, b}{\operatorname{argmin}} \sum_{i,j} max[1 - y_{ij}(b - (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{H}^T \mathbf{H} (\mathbf{x}_i - \mathbf{x}_j) + 2\mathbf{x}_i^T \mathbf{V}^T \mathbf{V} \mathbf{x}_j), 0] \tag{4.8}$$

where $b \in \mathbb{R}$ is a threshold, and y_{ij} is the label of a pair: $y_{ij} = 1$ if person i and j are the same and $y_{ij} = -1$, otherwise. For simplification, we denote $(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{H}^T \mathbf{H} (\mathbf{x}_i - \mathbf{x}_j) - 2\mathbf{x}_i^T \mathbf{V}^T \mathbf{V} \mathbf{x}_j$ as $d_{\mathbf{H},\mathbf{V}}(\mathbf{x}_i, \mathbf{x}_j)$. In addition, $\mathbf{H}$, $\mathbf{V}$, and b can be updated using a stochastic gradient descent algorithm as follows and are equally trained on positive and negative

pairs in turn:

$$\mathbf{H}_{t+1} = \begin{cases} \mathbf{H}_t, & \text{if } y_{ij}(b_t - d_{\mathbf{H},\mathbf{V}}(\mathbf{x}_i, \mathbf{x}_j)) > 1 \\ \mathbf{H}_t - \gamma y_{ij} \mathbf{H}_t \mathbf{\Psi}_{ij}, & \text{otherwise,} \end{cases}$$

$$\mathbf{V}_{t+1} = \begin{cases} \mathbf{V}_t, & \text{if } y_{ij}(b_t - d_{\mathbf{H},\mathbf{V}}(\mathbf{x}_i, \mathbf{x}_j)) > 1 \\ \mathbf{V}_t + \gamma y_{ij} \mathbf{V}_t \mathbf{\Gamma}_{ij}, & \text{otherwise,} \end{cases} \tag{4.9}$$

$$b_{t+1} = \begin{cases} b_t, & \text{if } y_{ij}(b_t - d_{\mathbf{H},\mathbf{V}}(\mathbf{x}_i, \mathbf{x}_j)) > 1 \\ b_t + \gamma_b y_{ij}, & \text{otherwise,} \end{cases}$$

where $\mathbf{\Psi}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T$, $\mathbf{\Gamma}_{ij} = \mathbf{x}_i \mathbf{x}_j^T + \mathbf{x}_j \mathbf{x}_i^T$, and γ is the learning rate for $\mathbf{H}$ and $\mathbf{V}$, and γ_b for the bias b. We perform whitening PCA to the extracted features and initialize both $\mathbf{H}$ and $\mathbf{V}$ with r largest eigenvectors. Note that $\mathbf{H}$ and $\mathbf{V}$ are updated only when the constraints are violated. The training and testing algorithms are summarized in Algorith 1 and Algorithm 2, respectively.

4.3 EXPERIMENTAL RESULTS

We present face verification results using the receiver operating characteristic (ROC) curves on three public datasets for unconstrained video-based face recognition: (1) Point-and-Shoot Challenge (PaSC) [7], (2) Multiple Biometric Grand Challenge (MBGC) [111], and (3) Face and Ocular Challenge Series (FOCS) [100].

Algorithm 1 LFVR TRAIN

Input: (1) Training images and labels for positive and negative pairs from LFW dataset [105] (2) patch size around each landmark, W_p, and (3) maxIter iterations.

Output: (1) Model parameters of Gaussians, $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, and w_i for $i = 1...K$, and (2) projection matrices learned from metric learning, $\mathbf{H}$ and $\mathbf{V}$.

1: Perform face and landmark detection for each training images.
2: Apply SQI to perform illumination normalization.
3: Extract multi-scale dense root-SIFT features from patches centered at each landmark and augment them with normalized x and y coordinates.
4: Learn $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, and w_i for each landmark $i = 1...K$ and fit a Gaussian using the mean and diagonal sample covariance of the extracted feature around ith landmarks as model parameters, and let each component share the same weight, $\frac{1}{K}$.
5: Perform FV encoding to the feature vectors.
6: Apply stochastic gradient descent using the training positive and negative face pairs in turn to optimize (4.8) until the maxIter iteration is reached to learn $\mathbf{H}$ and $\mathbf{V}$.

Algorithm 2 LFVR TEST

Input: (1) Model parameters of Gaussians, $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, and w_i for $i = 1...K$, (2) target and query videos, $\{\mathbf{T}\}_{i=1}^{N_t}$ and $\{\mathbf{Q}\}_{i=1}^{N_q}$, (3) projection matrices $\mathbf{H}$ and $\mathbf{V}$ to measure face similarity between a pair of images/videos, and (5) patch size around each landmark, W_p.

Output: Similarity matrix, $\mathbf{S}$.

1: Perform face detection and tracking for each target and query videos.
2: Perform landmark detection and align each face for all cropped faces of target and query videos.
3: Apply SQI to perform illumination normalization.
4: Extract multi-scale dense root-SIFT features from patches centered at each landmark and augment them with normalized x and y coordinates.
5: Aggregate the extracted features from each landmark and remove duplicates.
6: Perform FV encoding to feature vectors of frames of a video using the learned $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, and w_i for $i = 1...K$. and average all of them as the final descriptor.
7: Apply the learned joint Bayesian metric to each testing pair of faces to get the face similarity matrix, $\mathbf{S}$.

4.3.1 Implementation details

For preprocessing, we used OpenCV [14] and IVT [102] for face detection and face tracking respectively to perform face cropping. Then, we perform landmark detection using [29] [106] and perform similarity transform for face alignment. The image resolution after alignment is 63 $\times$ 80 pixels, and the distance between the centers of two eyes is about 10 pixels. The popular LFW still-face dataset and its label data (*i.e.* same pairs and different pairs) are used to learn the GMM and similarity measure. In addition, root-SIFT feature descriptors [112] are extracted using 16 $\times$ 16-pixel patches with 1-pixel stride on the face image. We repeat the extraction process at three scales with a scaling factor of $\sqrt{2}$. Before GMM learning, the features are first decorrelated by PCA to satisfy the diagonal covariance assumption of the GMM. When training the Gaussians, we aggregate the root-SIFT features with a 8 $\times$ 8-pixel patch centered at a landmark and use the mean and diagonal sample covariance of them as the Gaussian model parameters for the landmark. We also augment the training set with mirrored images. In testing stage, we aggregate the root-SIFT features within a 16 $\times$ 16-pixel patch centered at each landmark for error tolerance, and apply the FV encoding of the union of all features from the patches of all the landmarks. For better performance, the improved Fisher vector (IFV) [108] is used here, and the IFV is obtained by applying signed square-rooting and L_2 normalization steps. Finally, the LFVR representation is of dimension 12740 where K = 49 and d = 130. On the other hand, after applying the projection matrices, $\mathbf{H}$ and $\mathbf{V}$, learned from joint Bayesian metric learning, the dimensionality of the features reduces to 128 $\times$ 2 = 256. In addition, we take the average of all the FVs extracted from each frame of a video

as our video descriptor.

4.3.2 Point-and-Shoot Challenge

The PaSC is an evaluation challenge with 1401 videos of 265 people acquired with handheld cameras when subjects are involved in activities with non-frontal head pose and different illumination conditions. There are two types of experiments: (1) video-to-video and (2) still-to-video. In the video-to-video experiment, a person in a query video is compared to a set of target videos. Both target and query videos are from the same pool of 1401 videos. For the still-to-video experiment, the person in a query video is to be compared with a large set of still face images (4688 face image in total). The performance is evaluated with face verification at a false accept rate of 0.01 and the associated ROC curves.

The handheld videos consist of 1401 videos of 265 people acquired at the University of Notre Dame using five different handheld video cameras. The resolution for the videos ranges from 640×480 to 1280×720. Videos are acquired at six locations for a combination of different indoor and outdoor settings. The sample frames are shown in Fig. 4.4 which shows the challenging conditions due to variations of pose and illumination conditions for the PaSC dataset. We present the performance results of our approach for both tasks in Table 4.1 and in Fig. 4.5. From the table and the ROC curves, we see that our approach achieves better performance at FAR=0.01 for both tasks as compared with the results reported on the IJCB competition [7] (*i.e.* our approach ranked 3rd as compared with the results in the recent FG competition [113] in May, 2015. The top performer

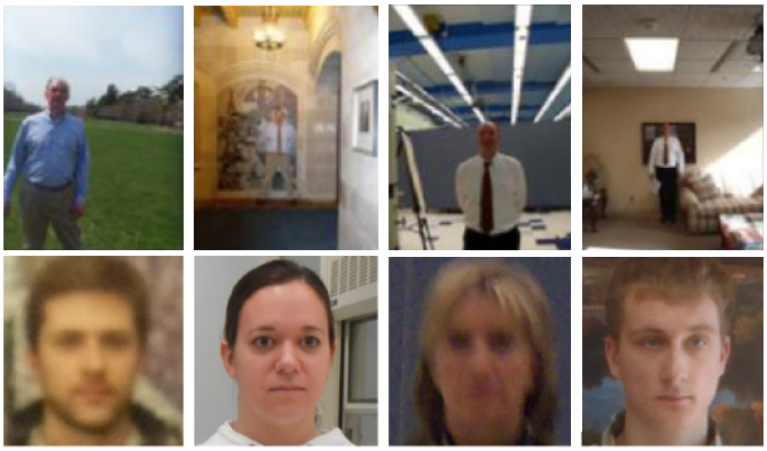
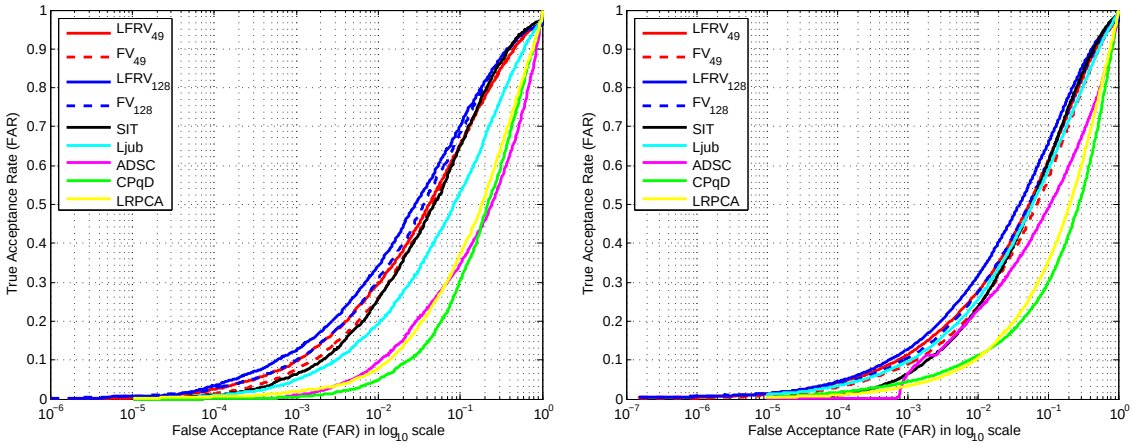


Figure 4.4: The upper row shows the sample frames of PaSC in four different scenarios. Each image/video is captured at different distance from camera. The last row shows the cropped face images are from still images of PaSC which demonstrate lighting, motion blur, and poor focus in point-and-shoot images.

used DCNN feature with manifold distance computed from image sets and achieved 0.59 verification rate at FAR=0.01 for the handheld video scenario. The method that placed second also used landmark-based feature vector based on a new texture descriptor, Dual-Cross Patterns [56], and they achieved 0.38). In Table 4.1, LFRV_{49} is used to denote the face that has 49 detected landmarks in our method. In addition, to boost the performance of our method, we also trained the GMM using the EM algorithm but only using the SIFT features within the regions surrounding to landmarks. We denote it as LFRV_{128}. We also denote the traditional FV trained using the EM with the features over the entire faces as FV_{49} and FV_{128} for 49 and 128 components, respectively. The learned GMMs for LFRV_{49}, FV_{49}, LFRV_{128}, and FV_{128} are illustrated in Fig. 4.3.

Group	Algorithm	Exp. 1	Exp. 2
ADSC	LBP-SIFT-WPCA-SILD	0.09	0.23
CpqD	ISV-GMM	0.05	0.11
SIT	Eigen-PEP	0.26	0.24
Ljub	PLDA-WPCA-LLR	0.19	0.26
CSU	LRPCA Baseline	0.08	0.10
Ours	FV_{49}	0.2583	0.2365
Ours	$LFVR_{49}$	0.2957	0.2749
Ours	FV_{128}	0.3095	0.2728
Ours	$LFVR_{128}$	0.3408	0.3152

Table 4.1: Face verification rates [7] at FAR = 0.01 for the unconstrained video-to-video (Exp. 1) and video-to-still (Exp. 2) tasks.



(a) (b)

Figure 4.5: (a) shows the ROC curves for the uncontrolled video-to-video face verification task of the PaSC dataset where the target and query videos are from the same set, and (b) shows the ROC curve for still-to-video task where still images are the target set and videos as query. The figure also shows our approach achieves better results at FAR=0.01 than previous state-of-the-art methods reported in IJCB 2014 competition for both tasks.

4.3.3 Multiple Biometric Grand Challenge

In the MBGC dataset, there are 146 subjects in total, and videos are available in two resolutions: standard definition (SD, 720 $\times$ 480 pixels) and high definition (HD, 1440 $\times$

1080 pixels). It consists of 399 walking sequences where 201 of them are in SD and 198 in HD, and 371 activity sequences where 185 in SD and 186 in HD. Fig. 4.7 shows the sample frames. For the walking sequences, subjects usually walk toward and keep their faces facing the camera for most of the time and turn their faces sideways at the end. The main challenge of the dataset comes from blur caused by motion, frontal and non-frontal faces with shadows which also lead to difficulty in tracking the faces in the video.

We also compare the verification results of the proposed method with DFRV [71] and the manifold-based method, WGCP [67]. These methods produced favorable results compared to several manifold and image set-based methods. As a result, we use them as the baseline algorithms. We perform verification experiments on the subsets of S2, S3, and S4 from the walking sequences where S2 is the set of subjects who have at least two face videos available, S3 at least three available, and S4 at least four available (S2: 144 subjects, 397 videos in total, S3: 55 subjects, 219 videos in total, and S4: 54 subjects, 216 videos). The verification results are shown in Fig. 4.6 and Table 4.2. It can be seen from this figure that the proposed approach achieves better results than DFRV, WGCP, and the one based on FV with the same number of components as our LFRV method. The results essentially demonstrate the effectiveness of dense multi-scale facial landmark features.

4.3.4 Face and Ocular Challenge Series

In addition to the MBGC dataset, we tested our approach on another challenging dataset, FOCS. The FOCS UT-Dallas dataset contains 510 walking and 506 activity video sequences for 295 subjects with the resolution, 720×480 pixels. The sample frames are

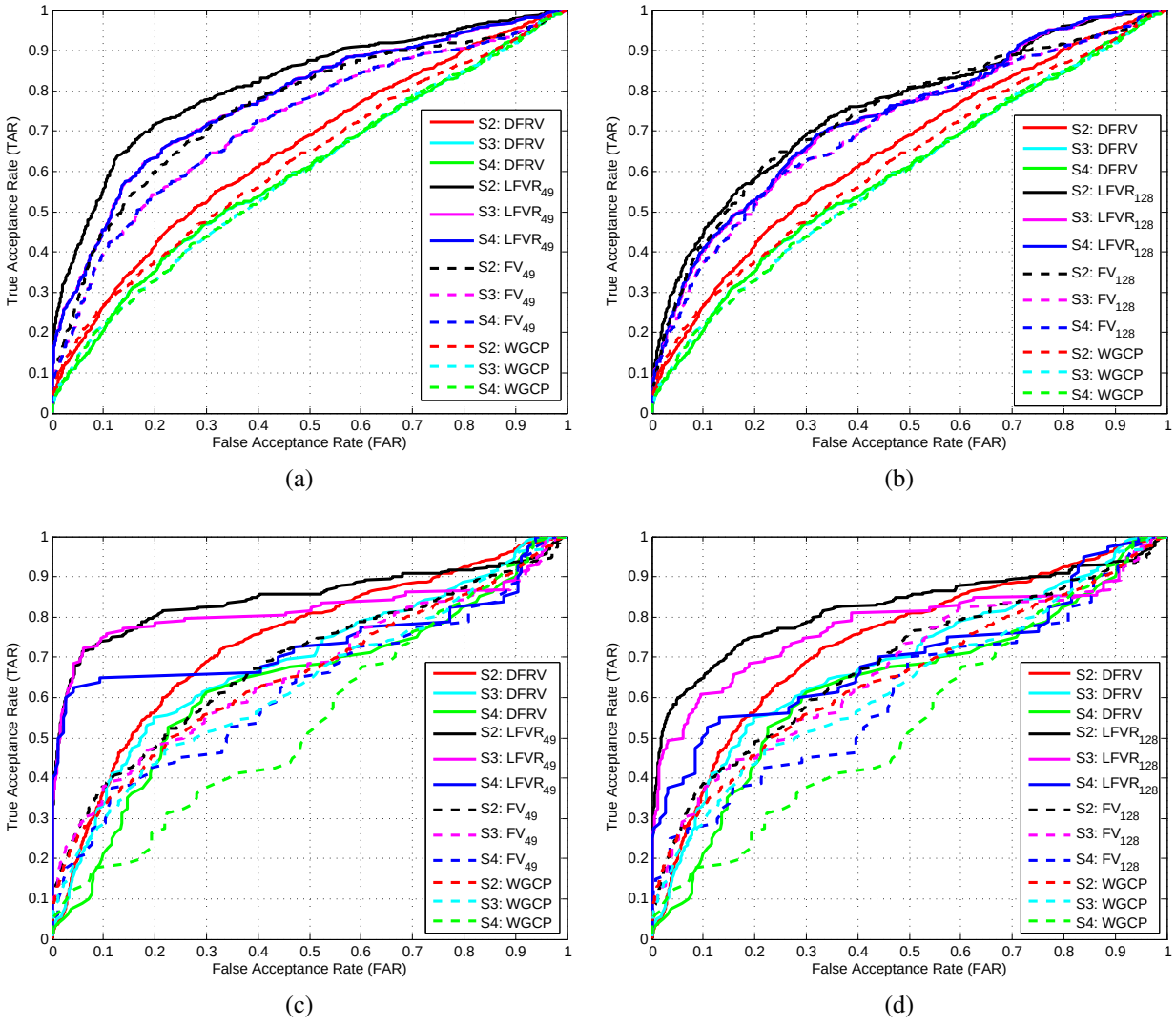


Figure 4.6: (a) and (b) show the ROC curves of face verification for subsets of S2, S3, and S4 for MBGC dataset where target and query videos are from the same set. (c) and (d) for the FOCS dataset. For these figures, we compare the results of LFVR of 49 (*i.e.* in (a)(c)) and 128 (*i.e.* in (b)(d)) components with DFRV and their FV counterparts using the same number of components respectively.

Figure 4.7: The upper row is the sample frames of MBGC walking sequences in four different scenarios, and the bottom row shows the sample frames from FOCS UT-Dallas walking videos.

shown in Fig. 4.7. The sequences were acquired on different days. For walking sequences, subjects initially stand far away from the camera, and then walk toward the camera while keeping their faces facing the camera and turn away at the end. We conducted the same verification tests as we did for MBGC subsets: S2 (189 subjects, 404 videos), S3 (19 subjects, 64 videos), and S4 (6 subjects, 25 videos) for UT-Dallas walking videos. The verification results are shown in Fig. 4.6 and Table 4.2. As in the MBGC case, the FOCS results also show that our proposed LFRV works more effectively than FV whose GMM is trained over the entire face. However, we can find from the results of both MBGC and FOCS that the performance of LFVR_{128} is worse than LFVR_{49}. One possible reason is that the resolution of detected faces in these two datasets is smaller than PaSC (*i.e.* about the half on average.) After alignment, the face images become blurred with fewer textural details. Thus, the performance saturated earlier when increasing the number of GMM components.

MBGC	WGCP [67]	DFRV [71]	FV_{49} [57]	FV_{128} [57]	$LFVR_{128}$ Ours	$LFVR_{49}$ Ours
S2	0.27	0.26	0.45	0.42	0.45	**0.58**
S3	0.22	0.22	0.40	0.38	0.40	**0.45**
S4	0.22	0.22	0.40	0.38	0.40	**0.45**
FOCS	WGCP	DFRV	FV_{49}	FV_{128}	$LFVR_{128}$	$LFVR_{49}$
S2	0.33	0.36	0.38	0.39	0.65	**0.74**
S3	0.29	0.34	0.38	0.37	0.61	**0.75**
S4	0.18	0.21	0.29	0.28	0.51	**0.65**

Table 4.2: it shows the verification rates of each algorithm at FAR=0.1. Our $LFVR_{49}$ achieves the best results.

4.4 Summary

In this chapter, we proposed a landmark-based Fisher vector representation for video-based face verification problems. Our experimental results demonstrate that if the landmarks are available, we should always utilize them. In addition, our approach greatly reduces the training time to learn a GMM and the dimensionality for the final feature representation while achieving better performance than the original Fisher vector counterpart.

Chapter 5: Unconstrained Still/Video-Based Face Verification with Deep Convolutional Neural Networks

5.1 Overview

Many algorithms have been shown to work well on images and videos that are collected in controlled settings. However, the performance of these algorithms often degrades significantly on images that have large variations in pose, illumination, expression, aging, and occlusion. In addition, for an automated face verification system to be effective, it also needs to handle errors that are introduced by algorithms for automatic face detection, face association, and facial landmark detection.

Existing methods have focused on learning robust and discriminative representations from face images and videos. One approach is to extract an over-complete and high-dimensional feature representation followed by a learned metric to project the feature vector onto low-dimensional space and then compute the similarity scores. For example, high-dimensional multi-scale local binary pattern (LBP) [55] features extracted from local patches around facial landmarks and Fisher vector (FV) [57] [114] features have been shown to be effective for face recognition. Despite significant progress, the performance of these systems has not been adequate for deployment. However, given the

availability of millions of annotated data, faster GPUs and a better understanding of the nonlinearities, DCNNs are providing much better performance on tasks such as object recognition [8] [9] [115], object/face detection [10] [24] [116] [117] [118], face verification/recognition [11] [63]. It has been shown that DCNN models can not only characterize large data variations but also learn a compact and discriminative representation when the size of the training data is sufficiently large. In addition, it can be generalized to other vision tasks by fine-tuning the pre-trained model on the new task [12].

In this chapter, we present an automated face verification system. Due to the robustness of DCNNs, we build each component of our system based on separate DCNN models. Modules for detection and face alignment use the DCNN architecture proposed in [8]. For face verification, we train two DCNN models trained using the CASIA-WebFace [13] dataset. Finally, we compare the performance of our approach with many face matchers on the IJB-A dataset which are being carried out or have been recently reported [119]. The proposed system is fully automatic and yields comparable or better performance than other existing algorithms when evaluated on IJB-A and CS2 datasets. Although the IJB-A dataset contains significant variations in pose, illumination, expression, resolution and occlusion which are much harder than the Labeled Faces in the Wild (LFW) datasets, we present verification results for the LFW dataset too.

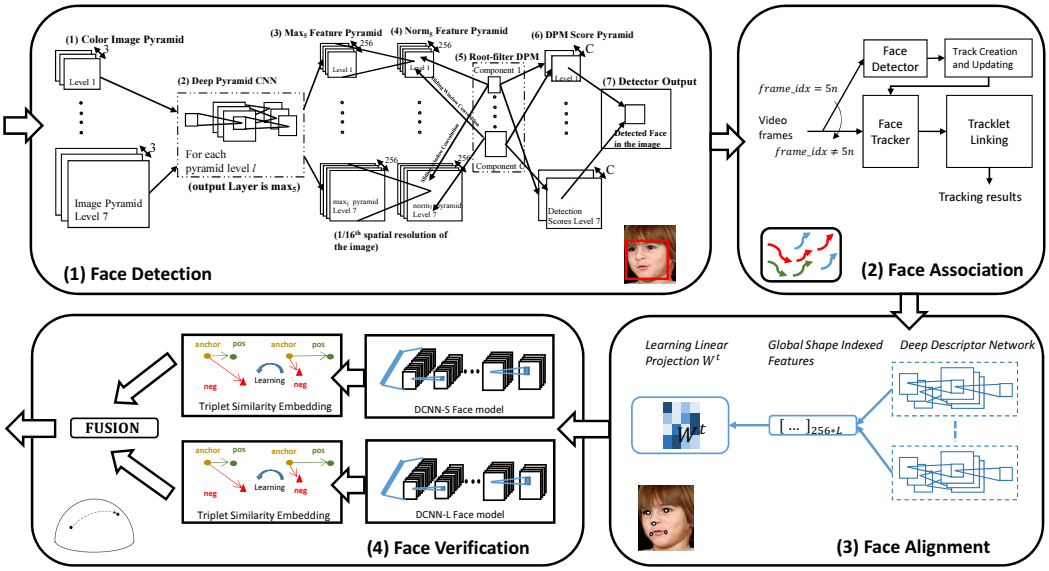


Figure 5.1: An overview of the proposed DCNN-based face verification system.

5.2 Proposed Approach

The proposed system is a complete pipeline for performing automatic face verification. Given a still image or a video, we first pass it through the face preprocessing modules: (1) face detection to localize faces in each image and video frame, (2) we associate the detected faces with the common identity for videos and (3) align the faces into canonical coordinates using the detected landmarks. Finally, we perform face verification to compute the similarity between a pair of images/videos. The system is illustrated in Figure 6.2. The details of each component are presented in the following sections.

5.2.1 Face Preprocessing

In this subsection, we introduce each face preprocessing modules used in this capture as follows.

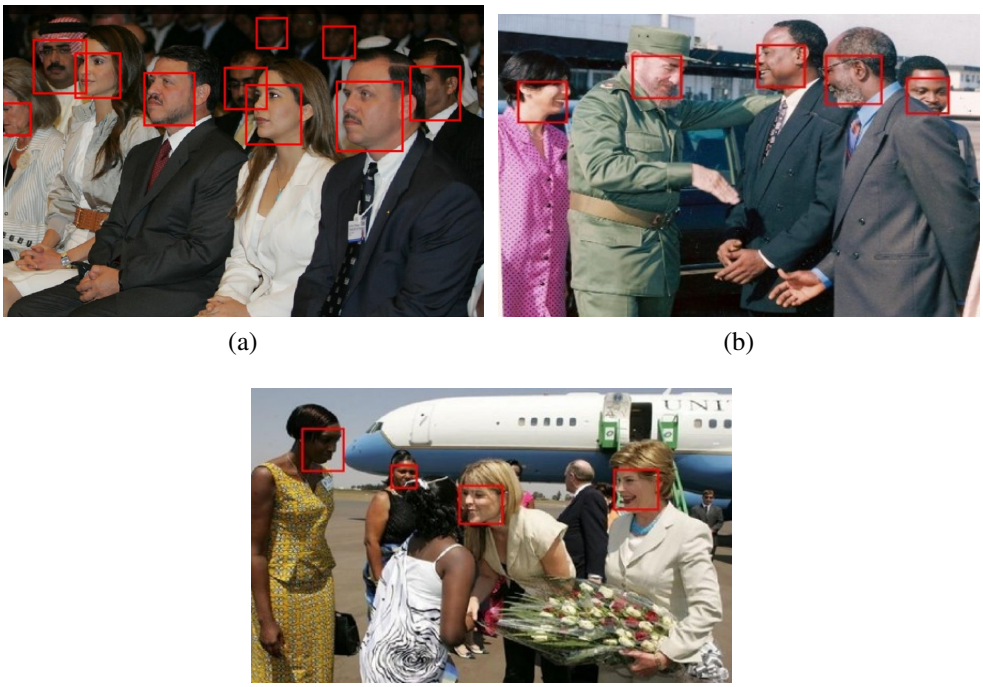
5.2.1.1 Face Detection

All the faces in the images/video frames are detected using a DCNN-based face detector, called the Deep Pyramid Deformable Parts Model for Face Detection (DP2MFD) [24], which consists of two modules. The first module generates a seven level normalized deep feature pyramid for any input image of arbitrary size, as illustrated in the first part of Figure 6.2. The same CNN architecture as Alexnet [8] is adopted for extracting the deep features. This image pyramid network generates a pyramid of 256 feature maps at the fifth convolution layer (conv_5). A 3×3 max filter is applied to the feature pyramid at a stride of one to obtain the max_5 layer. Typically, the activation magnitude for a face region decreases with the size of the pyramid level. As a result, a large face detected by a fixed-size sliding window at a lower pyramid level will have a high detection score compared to a small face getting detected at a higher pyramid level. In order to reduce this bias to face size, we apply a z-score normalization step on the max_5 features at each level. For a 256-dimensional feature vector $x_{i,j,k}$ at the pyramid level *i* and location (j, k), the normalized feature $x_{i,j,k}$ is computed as:

$$x_{i,j,k} = \frac{x_{i,j,k} - \mu_i}{\sigma_i}, \tag{5.1}$$

where μ_i is the mean feature vector, and σ_i is the standard deviation for the pyramid level *i*. We refer to the normalized max_5 features as $norm_5$. Then, the fixed-length features from each location in the pyramid are extracted using the sliding window approach.

The second module is a linear SVM, which takes these features as inputs to classify



(a) (b)

(c)

Figure 5.2: Sample detection results on an IJB-A image using the deep pyramid method.

each location as face or non-face, based on their scores. A root-only DPM is trained on the norm_5 feature pyramid using a linear SVM. In addition, the deep pyramid features are robust to not only pose and illumination variations but also to different scales. The DP2MFD algorithm works well in unconstrained settings as shown in Figure 5.2. We also present the face detection performance results under the face detection protocol of the IJB-A dataset in Section 5.3.

5.2.1.2 Facial Landmark Detection

Once the faces are detected, we perform facial landmark detection for face alignment. The proposed facial landmark detection algorithm, local deep descriptor regression (LDDR) [1], works in two stages. We model the task as a regression problem, where

beginning with the initial mean shape, the target shape is reached through regression. The first step is to perform feature extraction of a patch around a point of the shape followed by linear regression as described in [120] [30]. Given a face image I and the initial shape S^0, the regressor computes the shape increment ΔS from the deep descriptors and updates the face shape using (5.2).

$$S^t = S^{t-1} + W^t \Phi^t(I, S^{t-1}) \tag{5.2}$$

The CNN features (represented as Φ in 5.2) carefully designed with the proper number of strides and pooling (refer to Table 5.1 for more details), are used as the features to perform regression. We use the same CNN architecture as Alexnet [8] with the pretrained weights for the ImageNet dataset as shown in Figure 5.3. Then, we further fine-tuned it with AFLW [121] dataset for face detection task. The fine-tuning step helps the network to learn features specific to faces. Furthermore, we adopt the cascade regression, in which the output generated by the first stage is used as an input for the next stage. The number of stages is fixed at 5 in our system. The patches selected for feature extraction are reduced subsequently in later stages to improve the localization of facial landmarks. After the facial landmark detection is completed, each face is aligned into the canonical coordinate using the similarity transform and seven landmark points (*i.e.*, two left eye corners, two right eye corners, nose tip, and two mouth corners).

Stage 1	Input Size (pixels)	conv1	max1	conv2	max2
Stage 1	92×92	4	2	1	1
Stage 2	68×68	3	2	1	1
Stage 3	42×42	2	1	1	2
Stage 4	21×21	1	1	1	1

Table 5.1: Input size and the number of strides in conv1, max1, conv2 and max2 layers for 4 stages of regression.

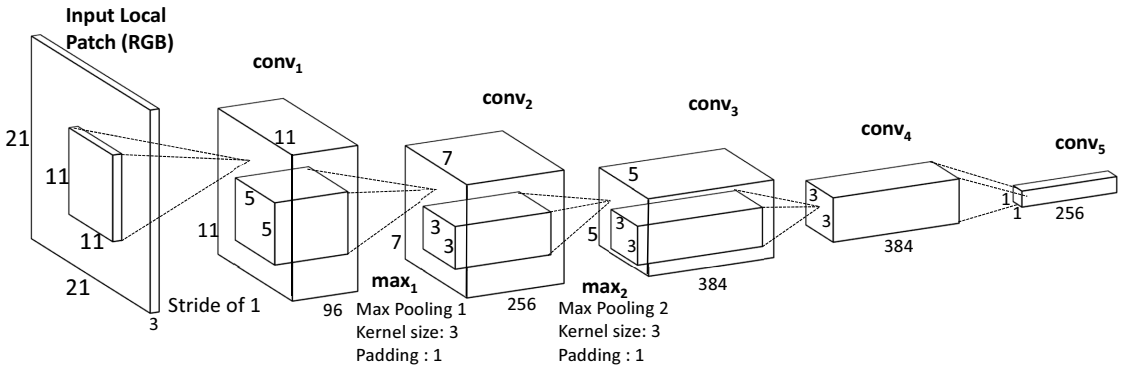


Figure 5.3: The DCNN architecture used to extract the local descriptors for the facial landmark detection task [1].

5.2.1.3 Face Association

Because there are multiple subjects appearing in the frames of each video of the IJB-A dataset, performing face association to assign each face to its corresponding subject is an important step to pick the correct subject for face verification. Thus, once the faces in the images and video frames are detected, we track multiple faces by integrating results from the face detector, face tracker, and a tracklet linking step. The second part of Figure 6.2 shows the block diagram of the multiple face tracking system. We apply the face detection algorithm in every fifth frame using the face detection method presented in

Section 5.2.1.1. The detected bounding box is considered as a novel detection if it does not have an overlap ratio with any bounding box in the previous frames larger than γ. The overlap ratio of a detected bounding box $\mathbf{b}_d$ and a bounding box $\mathbf{b}_{tr}$ in the previous frames is defined as

$$s(\mathbf{b}_d, \mathbf{b}_{tr}) = \frac{area(\mathbf{b}_d \cap \mathbf{b}_{tr})}{area(\mathbf{b}_{tr})}. \tag{5.3}$$

We empirically set the overlap threshold γ to 0.2. A face tracker is created from a detection bounding box that is treated as a novel detection. We set the face detection confidence threshold to -1.0 to select the bounding boxes of face detection of high confidence. For face tracking, we use the Kanade-Lucas-Tomasi (KLT) feature tracker [44] to track the faces between two consecutive frames. To avoid the potential drift of trackers, we update the bounding boxes of the tracker by those provided by the face detector in every fifth frame. The detection bounding box $\mathbf{b}_d$ replaces the tracking bounding boxes $\mathbf{b}_{tr}$ of a tracklet in the previous frame if $s(\mathbf{b}_d, \mathbf{b}_{tr}) \leq \gamma$. A face tracker is terminated if there is no corresponding face detection overlapping with it for more than t frames. We set t to 4 based on empirical grounds.

In order to handle the fragmented face tracks resulting from occlusions or unreliable face detection, we use the tracklet linking method proposed by [39] to associate the bounding boxes in the current frames with tracklets in the previous frames. The tracklet linking method consists of two stages. The first stage is to associate the bounding boxes provided by the tracker or the detector in the current frame with the existing tracklet in previous frames. This stage consists of local and global associations. The local association step associates the bounding boxes with the set of tracklets, having high confidence.

Figure 5.4: Sample results of our face association method for videos of JANUS CS2 which is the extension dataset of IJB-A.

The global step associates the remaining bounding boxes with the set of tracklets of low confidence. The second stage is to update the confidence of the tracklets, which will be used for determining the tracklets for local or global association in the first stage. We show sample face association results for some videos from the CS2 dataset in Figure 5.4.

5.2.2 Face Verification based on Deep Convolutional Neural Networks

After face preprocessing, we come to our core modules to perform the face verification task which is based on deep convolutional neural network. We give the details for learning the representation and similarity measure as follows.

5.2.2.1 Deep Convolutional Face Representation

In this chapter, we train two deep convolutional networks. One is trained using tight face bounding boxes (DCNN$_S$), and the other using large bounding boxes which include more contextual (DCNN$_L$) information. In Section 5.3, we present results which show that both networks capture discriminative information and complement each other. In ad-

dition, the fusion of two networks does significantly improve the final performance. The architectures of both networks are summarized in Tables 5.2.

Stacking small filters to approximate large filters and building very deep convolutional networks reduces the number of parameters but also increases the nonlinearity of the network in [122] [9]. In addition, the resulting feature representation is compact and discriminative. Therefore, for (DCNN$_S$), we use the same network architecture presented in [87] and train it using the CASIA-WebFace dataset [13]. The dimensionality of the input layer is $100 \times 100 \times 3$ for RGB images. The network includes ten convolutional layers, five pooling layers, and one fully connected layer. Each convolutional layer is followed by a parametric rectified linear unit (PReLU) [123], except the last one, conv52. Moreover, two local normalization layers are added after conv12 and conv22, respectively, to mitigate the effect of illumination variations. The kernel size of all filters is 3×3. The first four pooling layers use the max operator, and pool$_5$ uses average pooling. The feature dimensionality of pool$_5$ is thus equal to the number of channels of conv52 which is 320. The dropout ratio is set as 0.4 to regularize Fc6 due to the large number of parameters (*i.e.* 320×10548[1] .). The pool$_5$ feature is used for face representation. The extracted features are further L_2-normalized to unit length before the metric learning stage. If there are multiple images and frames available for the subject template, we use the average of pool$_5$ features as the overall feature representation.

[1]The list of overlapping subjects is available at `http://www.umiacs.umd.edu/~pullpull/janus_overlap.xlsx`

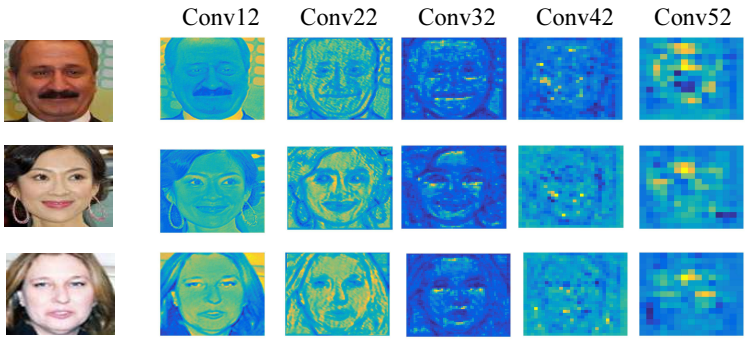


Figure 5.5: An illustration of some feature maps of conv12, conv22, conv32, conv42, and conv52 layers of DCNN$_S$ trained for the face identification task. At upper layers, the feature maps capture more global shape features which are also more robust to illumination changes than conv12. The feature maps are rescaled to the same size for visualization purpose. The green pixels represent high activation values, and the blue pixels represent low activation values as compared to the green.

On the other hand, for DCNN$_L$, the deep network architecture closely follows the architecture of the AlexNet [124] with some notable differences: reduced number of parameters in the fully connected layers; use of Parametric Rectifier Linear units (PReLU's) instead of ReLU, since they allow a negative value for the output based on a learnt threshold and have been shown to improve the convergence rate [123].

The reason for using the AlexNet architecure in the convolutional layers is due to the fact that we initialize the convolutional layer weights with weights from the AlexNet model which was trained using the ImageNet challenge dataset. Several recent works ([125], [126]) have empirically shown that this transfer of knowledge across different networks, albeit for a different objective, improves performance and more significantly reduces the need to train using a large number of iterations. To learn more domain specific information, we add an additional convolutional layer, *conv6* and initialize the fully connected layers *fc6-fc8* from scratch. Since the network is used as a feature extractor,

Name	Type	Filter Size/Stride	#Params
conv11	convolution	3×3 / 1	0.84K
conv12	convolution	3×3 / 1	18K
pool1	max pooling	2×2 / 2	
conv21	convolution	3×3 / 1	36K
conv22	convolution	3×3 / 1	72K
pool2	max pooling	2×2 / 2	
conv31	convolution	3×3 / 1	108K
conv32	convolution	3×3 / 1	162K
pool3	max pooling	2×2 / 2	
conv41	convolution	3×3 / 1	216K
conv42	convolution	3×3 / 1	288K
pool4	max pooling	2×2 / 2	
conv51	convolution	3×3 / 1	360K
conv52	convolution	3×3 / 1	450K
pool5	avg pooling	7×7 / 1	
dropout	dropout (40%)		
fc6	fully connected	10548	3296K
loss	softmax	10548	
total			5M

The architectures of DCNN_S.

Name	Type	Filter Size/Stride	#Params
conv1	convolution	11×11 / 4	35K
pool1	max pooling	3×3 / 2	
conv2	convolution	5×5 / 2	614K
pool2	max pooling	3×3 / 2	
conv3	convolution	3×3 / 2	885K
conv4	convolution	3×3 / 2	1.3M
conv5	convolution	3×3 / 1	885K
conv6	convolution	3×3 / 1	590K
pool6	max pooling	3×3 / 2	
fc6	fully connected	1024	9.4M
dropout	dropout (50%)		
fc7	fully connected	512	524K
dropout	dropout (50%)		
fc8	fully connected	10548	5.5M
loss	softmax	10548	
total			19.8M

The architecture of DCNN_L.

Table 5.2: The architecture for both DCNN_S and DCNN_L.

the last layer *fc8* is removed during deployment, thus reducing the number of parameters to 15M. When the network is deployed. the features are extracted from *fc7* layers resulting in a dimensionality of 512. The network is trained using the CASIA-WebFace dataset [13]. The dimensionality of the input layer is $227 \times 227 \times 3$ for RGB images.

In Figure 5.5, we show some feature activation maps of the DCNN_S model. At the upper layers, the feature maps capture more global shape features which are also more robust to illumination changes than Conv12 where the green pixels represent high activation values, and the blue pixels represent low activation values compared to the green.

5.2.2.2 Triplet Similarity Embedding

To further improve the performance of our deep features, we obtain a low-dimensional discriminative projection of the deep features, called the Triplet Similarity Embedding

(TSE) that is learnt using the training data provided for each split of IJB-A. The output of the procedure is an embedding matrix $\mathbf{W} \in \mathbf{R}^{n \times M}$ where M is the dimensionality of the deep descriptor (320 for DCNN_S and 512 for DCNN_L) and we set $n = 128$, thus achieving dimensionality reduction in addition to an improvement in performance.

In addition, for the triplet similarity embedding approach, the objective was two-fold (1) to achieve as small dimensionality as possible for both networks (2) to obtain a more discriminative representation in the low dimensional space which means to push similar pairs together and dissimilar pairs apart in the low-dimensional space. For learning $\mathbf{W}$, we solve an optimization problem based on constraints involving triplets - each containing two similar samples and one dissimilar sample. Consider a triplet $\{a, p, n\}$, where a (anchor) and p (positive) are from the same class, but n (negative) belongs to a different class. Our objective is to learn a linear projection $\mathbf{W}$ from the data such that the following constraint is satisfied:

$$(\mathbf{W}a)^T \cdot (\mathbf{W}p) > (\mathbf{W}a)^T \cdot (\mathbf{W}n) \tag{5.4}$$

In our case, $\{a, p, n\} \in \mathbf{R}^M$ are deep descriptors which are normalized to unit length. As such, $(\mathbf{W}a)^T \cdot (\mathbf{W}p)$ is the dot-product or the similarity between a, p under the projection $\mathbf{W}$. The constraint in (5.4) requires that the similarity between the anchor and positive samples should be higher than the similarity between the anchor and negative samples in the low dimensional space represented by $\mathbf{W}$. Thus, the mapping matrix $\mathbf{W}$ pushes similar pairs closer and dissimilar pairs apart, with respect to the anchor point. By choosing the dimensionality of $\mathbf{W}$ as $n \times M$ where $n < M$, we achieve dimensionality

reduction in addition to better performance. For our work, we fix $n = 128$ based on cross validation.

Given a set of labeled data points, we solve the following optimization problem:

$$\underset{\mathbf{W}}{\text{argmin}} \sum_{a,p,n \in \mathbb{T}} max(0, \alpha + a^T \mathbf{W^T W} n - a^T \mathbf{W^T W} p) \tag{5.5}$$

where $\mathbb{T}$ is the set of triplets and α is a margin parameter chosen based on the validation set. In practice, the above problem is solved in a Large-Margin framework using Stochastic Gradient Descent (SGD) and the triplets are sampled online. The update step for solving (5.5) with SGD is:

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta * \mathbf{W}_t * (a(n-p)^T + (n-p)a^T) \tag{5.6}$$

where $\mathbf{W}_t$ is the estimate at iteration t, $\mathbf{W}_{t+1}$ is the updated estimate, $\{a, p, n\}$ is the triplet sampled at the current iteration and η is the learning rate which is set to 0.01 for the current work.

More details regarding the optimization algorithm can be found in [127]. At each iteration, we sample 1000 instances from the whole training set to choose the negatives. Since the training set is relatively small for the datasets considered in this experiment, the entire training set is held in memory. Going forward this could be made efficient by using a buffer which will be replenished periodically, thus requiring a constant memory requirement. The computational complexity of each iteration is $O(M^2)$, that is, the complexity

varies quadratically with the dimension of the deep descriptor. The technique closest to the one presented in this section, which is used in recent works ([63], [11]) computes the embedding $\mathbf{W}$ based on satisfying the distance constraints given below:

$$\underset{\mathbf{W}}{\operatorname{argmin}} \sum_{a,p,n \in \mathbb{T}} max\{0, \alpha + (a-p)^T \mathbf{W}^\mathbf{T}\mathbf{W}(a-p) - \tag{5.7}$$

$$(a-n)^T \mathbf{W}^\mathbf{T}\mathbf{W}(a-n)\} \tag{5.8}$$

To be consistent with the terminology used in this chapter, we call it Triplet Distance Embedding (**TDE**). It should be noted that the **TSE** formulation is different from **TDE**, in that, the current work uses inner-product based constraints between triplets to optimize for the embedding matrix as opposed to norm-based constraints used in the **TDE** method. To choose the dimensionality, we test the values 64,128,256 using a 5 fold validation scheme for each split. The learning rate is chosen as 0.02 and is fixed throughout the procedure. The margin parameter is chosen as 0.1. We find from our experiments that lower margin works better but since we perform hard negative mining at each step, the method is not particularly sensitive to the margin parameter.

In general, to learn a reasonable distance measure directly using pairwise or triplet metric learning approach requires huge amount of data (*i.e.,*, the state-of-the-art approach [11] uses 200M images). In addition, the proposed approach decouples the DCNN feature learning and metric learning due to memory constraints. To learn a reasonable distance measure requires generating informative pairs or triplets. The batch size used for SGD is limited by the memory size of the graphics card. If the model is trained end-to-end, then

only a small batch size is available for use. Thus, in this work, we perform DCNN model training and metric learning independently. In addition, for the publicly available deep model [63], it is also trained first with softmax loss and followed by finetuning the model with verification loss with freezing the convolutional and fully connected layers except the last one to learn the transformation which is equivalent to the proposed approach.

5.3 Experimental Results

In this section, we present the results of the proposed automatic system for both face detection and face verification tasks on the challenging IARPA Janus Benchmark A (IJB-A) [128], its extended version Janus Challenging set 2 (JANUS CS2) dataset, and the LFW dataset. The JANUS CS2 dataset contains not only the sampled frames and images in the IJB-A, but also the original videos. In addition, the JANUS CS2 dataset[2] includes considerably more test data for identification and verification problems in the defined protocols than the IJB-A dataset. The receiver operating characteristic curves (ROC) and the cumulative match characteristic (CMC) scores are used to evaluate the performance of different algorithms for face verification. The ROC curve measures the performance in the verification scenarios, and the CMC score measures the accuracy in closed set identification scenarios.

5.3.1 Face Detection on IJB-A

The IJB-A dataset contains images and sampled video frames from 500 subjects collected from online media [128], [129]. For the face detection task, there are 67,183

[2]The JANUS CS2 dataset is not publicly available yet.

faces of which 13,741 are from images and the remaining are from videos. The locations of all faces in the IJB-A dataset have been manually annotated. The subjects were captured so that the dataset contains wide geographic distribution. Nine different face detection algorithms were evaluated on the IJB-A dataset [129], and the algorithms compared in [129] include one commercial off the shelf (COTS) algorithm, three government off the shelf (GOTS) algorithms, two open source face detection algorithms (OpenCV's Viola Jones and the detector provided in the Dlib library), and GOTS ver 4 and 5. In Figure 5.7, we show the precision-recall (PR) curves and the ROC curves, respectively corresponding to the method used in our work and one of the best reported methods in [129]. From the results, we see that the face detection algorithm used in our system outperforms the best performing method reported in [129] by a large margin. In Figure 5.8 (b), we illustrate typical faces in the IJB-A dataset that are not detected by DP2MFD, and we can find the faces to be usually in very extreme conditions which contain limited information for face verification. However, in Figure 5.8 (a), we also show that the DP2MFD algorithm can handle very difficult faces but relatively reasonable as compared to those in 5.8 (b). As shown in Figure 5.6, our DP2MFD algorithm also achieves top performance in the challenging FDDB benchmark [130] for face detection with a large performance margin compared to most algorithms. Some of the recent published methods compared in the FDDB evaluation include Faceness [20], HeadHunter [19], JointCascade [15], CCF [131], Squares- ChnFtrs-5 [19], CascadeCNN [17], Structured Models [132], DDFD [21], NDP-Face [133], PEP-Adapt [134] and TSM [135]. More comparison results with other face detection data sets are available in [24]. Since the CS2 dataset has not been released to public, we are not able to provide comparisons with other existing face detectors.

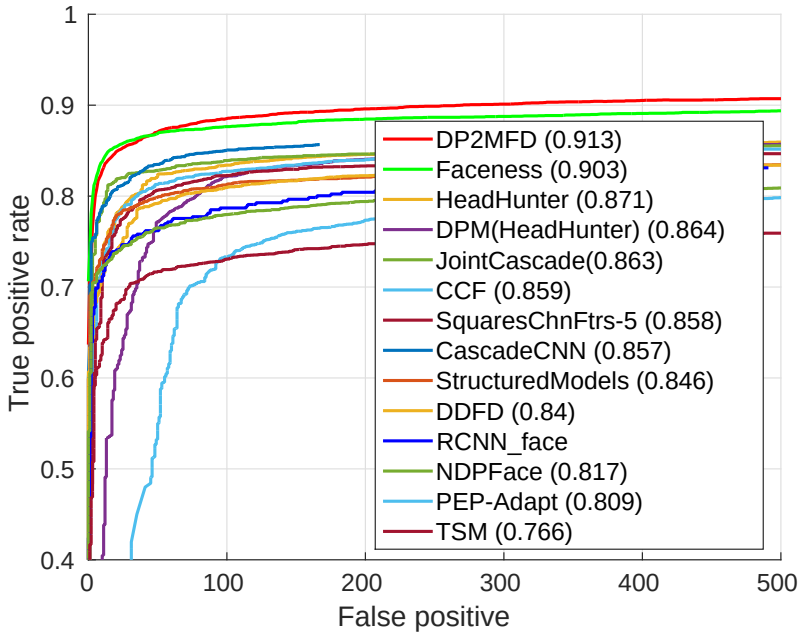
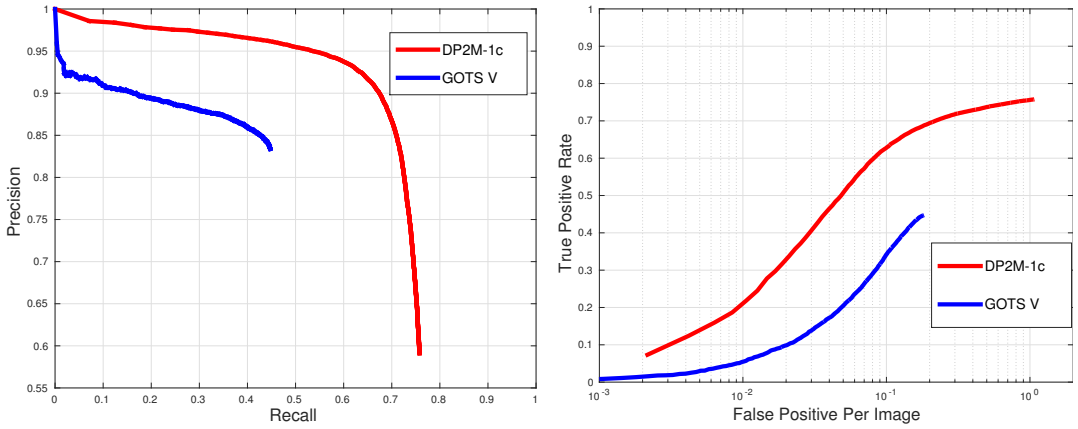


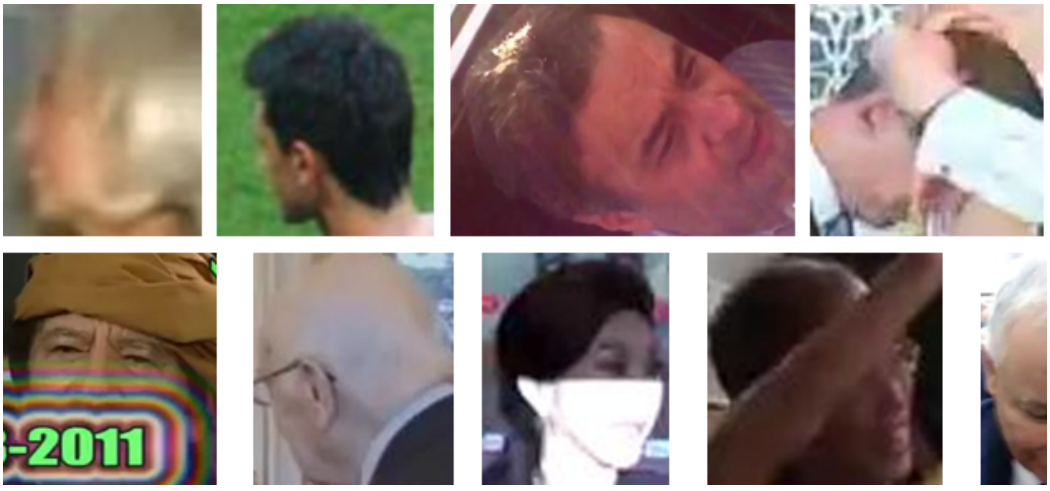
Figure 5.6: Face detection performance evaluation on the FDDB dataset.



(a) (b)

Figure 5.7: Face detection performance evaluation on the IJB-A dataset. (a) Precision vs. recall curves. (b) ROC curves.

(a)



(b)

Figure 5.8: (a) shows the difficult faces in the IJB-A dataset that are successfully detected by DP2MFD, and (b) shows faces that are not detected by DP2MFD. From the results, we can see that DP2MFD can handle difficult occlusion, partial face, large illumination and pose variations.

5.3.2 Facial Landmark Detection on IJB-A

In this section, we evaluate the performance of the facial landmark detection method used in this work on the IJB-A dataset for the performance evaluation. For the training data, we take 3148 images in total from the LFPW [136], Helen [137] and AFW [135] datasets and test on IJBA-A dataset. The subjects were captured so that the dataset contains wide geographic distribution. The challenge comes through the wide diversity in pose, illumination and resolution. Our method produce 68 facial landmark points following MultiPIE [138] markup format. We evaluate the performance using the Normalized Mean Square Error and average pt-pt error (normalized by face size) vs fraction of images plots of different methods. Since IJB-A is annotated only with 3 key-points on the faces (two eyes and nose base) by human annotators, the interoccular distance error was normalized by the distance between nose tip and the midpoint of the eye centers. In Figure 5.9, we present a comparison of our algorithm with [135], [139] and [140]. For the Helen dataset, we show the performance of 49-point and full 68-point results in Table 5.3. Our deep descriptor-based global shape regression method outperforms the above mentioned state-of-the-art methods in both high-quality (Helen) and low-quality (IJB-A) images. Samples detected landmarks results are shown in Figure 5.10. More evaluation results for landmark detection other standard data sets may be found [1]. Once the facial landmark detection is completed, we choose seven landmark points (*i.e.* two left eye corners, two right eye corners, nose tip, and two mouth corners) out of the detected 68 points and apply the similarity transform to warp the faces into canonical coordinates.

Method	*68-pts*	*49-pts*
Zhu *et al. [135]*	8.16	7.43
DRMF [139]	6.70	-
RCPR [141]	5.93	4.64
SDM [142]	5.50	4.25
GN-DPM [143]	5.69	4.06
CFAN [144]	5.53	-
CFSS [145]	4.63	3.47
LDDR	**4.76**	**2.36**

Table 5.3: Averaged error comparison of different methods on the Helen dataset.

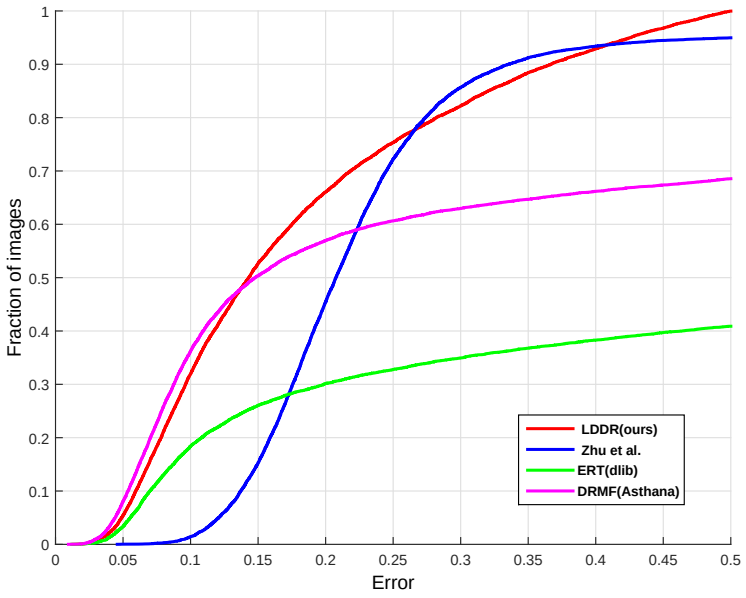


Figure 5.9: Average 3-pt error (normalized by eye-nose distance) vs fraction of images in the IJB-A dataset.

Figure 5.10: Sample facial landmark detection results.

5.3.3 IJB-A and JANUS CS2 for Face Verification

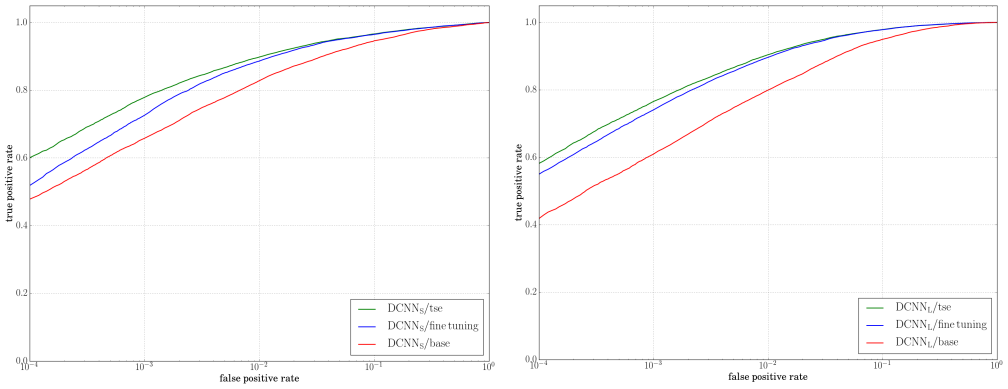
For the face verification task, both IJB-A and JANUS CS2 datasets contain 500 subjects with 5,397 images and 2,042 videos split into 20,412 frames, 11.4 images and 4.2 videos per subject. Sample images and video frames from the datasets are shown in Figure 5.11. (*i.e.*, the videos are only released for the JANUS CS2 dataset.) The IJB-A evaluation protocol consists of verification (1:1 matching) over 10 splits. Each split contains around 11,748 pairs of templates (1,756 positive and 9,992 negative pairs) on average. Similarly, the identification (1:N search) protocol also consists of 10 splits, which are used to evaluate the search performance. In each search split, there are about 112 gallery templates and 1,763 probe templates (*i.e.* 1,187 genuine probe templates and 576 impostor probe templates). On the other hand, for the JANUS CS2, there are about 167 gallery templates and 1,763 probe templates and all of them are used for both identification and verification. The training set for both datasets contains 333 subjects, and

the test set contains 167 subjects without any overlapping subjects. Ten random splits of training and testing are provided by each benchmark, respectively. The main differences between IJB-A and JANUS CS2 evaluation protocols are that (1) IJB-A considers the open-set identification problem and the JANUS CS2 considers the closed-set identification and (2) IJB-A considers the more difficult pairs which are the subsets from the JANUS CS2 dataset.

Figure 5.11: Sample images and frames from the IJB-A (top) and JANUS CS2 datasets (bottom). Challenging variations due to pose, illumination, resolution, occlusion, and image quality are present in these images.

Unlike the LFW and YTF datasets, which only use a sparse set of negative pairs to evaluate the verification performance, the IJB-A and JANUS CS2 both divide the images/video frames into gallery and probe sets so that all the available positive and negative pairs are used for the evaluation. Also, each gallery and probe set consist of multiple templates. Each template contains a combination of images or frames sampled from multiple image sets or videos of a subject. For example, the size of the similarity matrix for JANUS CS2 split1 is 167×1806 where 167 are for the gallery set and 1806 for the probe set (*i.e.* the same subject reappears multiple times in different probe templates). Moreover, some templates contain only one profile face with a challenging pose with low quality imagery. In contrast to LFW and YTF datasets, which only include faces detected by the Viola

Jones face detector [14], the images in the IJB-A and JANUS CS2 contain extreme pose, illumination, and expression variations. These factors essentially make the IJB-A and JANUS CS2 challenging face recognition datasets [128].



(a) (b)

Figure 5.12: The performance evaluation for face verification tasks of (a) DCNN$_S$ and (b) DCNN$_L$ of before finetuning, with finetuning, and with finetuning and triplet similarity embedding for the JANUS CS2 dataset under Setup 3 (semi-automatic mode). Fine tuning is done only using the training data in each split.

5.3.4 Performance Evaluations of Face Verification on IJB-A and JANUS CS2

To take different situations into account, we have considered three modes of evaluations, manual, automatic and semi-automatic modes. This enables the handling of cases where we are unable to detect any of the faces (*i.e.,* the failure of face detection.) in the images of the given template and also to compare the performance with the one using the metadata provided with the dataset. We describe the setups of performance evaluation in details as follows:

- **Setup 1 (manual mode):** Under this setup, we directly use the three facial land-

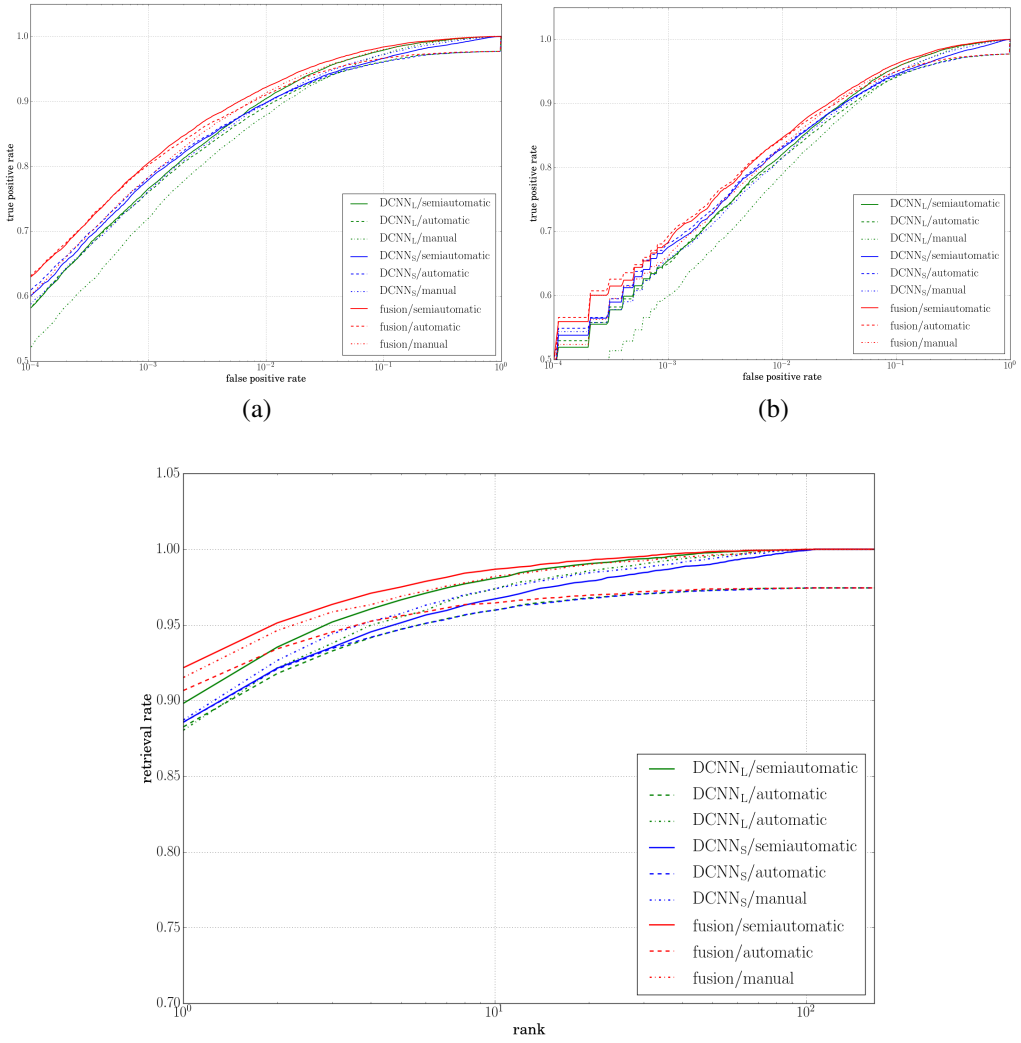


Figure 5.13: (a) and (b) show the face verification performance of the fusion model for JANUS CS2 and IJB-A (1:1) verification, respectively, and (c) shows the face identification performance of the fusion model for IJB-A (1:N) identification for all the three setups. Fine tuning is done only using the training data in each split.

marks and face bounding boxes provided along with the datasets.

- **Setup 2 (automatic mode):** In this setup when we get a video we use the face association method to detect and track the faces and to extract the bounding box to perform fiducial detection. If it is an image, we perform detection and facial landmark detection independently. For every image or frame in a template in which we are unable to detect the target person, we are unable to compare the template with others and thus assign all the corresponding entries for the template in the similarity matrices to the lowest similarity scores, -Inf.

- **Setup 3 (semi-automatic mode):** In this setup if we are able to detect the target person in an image then we follow setup 2. Otherwise, we follow setup 1 to use the metadata of the dataset for the faces which are not detected and tracked by our algorithms.

To evaluate the performance of two networks individually, we present the ROC curves of DCNN_S and DCNN_L of the Setup 3 (*i.e.*, semi-automatic mode) for the JANUS CS2 dataset in Figure 5.12. As shown in the figures, the performances are consistently improved for both networks after fine-tuning the models previously trained using CASIA-WebFace dataset on the training data of JANUS CS2. Triplet similarity embedding (TSE) further increase the performance for both networks, especially for the TAR number at the low FAR interval. For all the results presented here, fine tuning is done using only the training data in each split. The gallery dataset is not used for parameter finetuning or for triplet similarity embedding. Then, we perform the fusion of the two networks by adding the corresponding similarity scores together and demonstrate the fusion results of

all the three setup for the verification task of both JANUS CS2 and IJB-A in Figure 5.13 (a) and (b), respectively. In Figure 5.13 (c), we present the CMC curve for the IJB-A identification task. From Figure 5.13, it can be seen that even the simple fusion strategy used in this work significantly boosts the performance. Since DCNN_S is trained using tight face bounding boxes (DCNN_S) and DCNN_L using the large ones which includes more context (DCNN_L), one possible reason for the performance improvement is that the two networks contain discriminative information learned from different scales and complement each other. In addition, the figure also shows that the performance of our system in Setup 2 (the automatic mode) is comparable to Setup 1 (the manual mode) and Setup 3 (the semi-automatic mode). This demonstrates the robustness of each component of our system.

IJB-A-Verif	DCNN (setup 1)	DCNN (setup 2)	DCNN (setup 3)	DCNN_m (setup 1)	DCNN_m (setup 2)	DCNN_m (setup 3)
FAR=1e-2	0.834 ± 0.036	0.844 ± 0.026	0.846 ± 0.029	0.863 ± 0.02	0.885 ± 0.014	$\mathbf{0.889 \pm 0.016}$
FAR=1e-1	0.956 ± 0.008	0.95 ± 0.005	0.962 ± 0.007	0.966 ± 0.05	0.954 ± 0.003	$\mathbf{0.968 \pm 0.005}$
IJB-A-Ident	DCNN (setup 1)	DCNN (setup 2)	DCNN (setup 3)	DCNN_m (setup 1)	DCNN_m (setup 2)	DCNN_m (setup 3)
Rank-1	0.915 ± 0.011	0.907 ± 0.011	0.922 ± 0.011	0.916 ± 0.009	0.923 ± 0.01	$\mathbf{0.942 \pm 0.008}$
Rank-5	0.969 ± 0.007	0.955 ± 0.007	0.975 ± 0.006	0.971 ± 0.007	0.961 ± 0.006	$\mathbf{0.98 \pm 0.005}$
Rank-10	0.982 ± 0.005	0.965 ± 0.005	0.987 ± 0.001	0.981 ± 0.005	0.969 ± 0.004	$\mathbf{0.988 \pm 0.003}$
IJB-A-Ident	DCNN (setup 1)	DCNN (setup 2)	DCNN (setup 3)	DCNN_m (setup 1)	DCNN_m (setup 2)	DCNN_m (setup 3)
FPIR=0.01	0.618 ± 0.05	0.64 ± 0.043	0.631 ± 0.041	0.639 ± 0.057	0.646 ± 0.055	$\mathbf{0.654 \pm 0.001}$
FPIR=0.1	0.799 ± 0.014	0.806 ± 0.012	0.813 ± 0.014	0.816 ± 0.015	0.827 ± 0.012	$\mathbf{0.836 \pm 0.01}$

Table 5.4: Results on the IJB-A dataset. The TAR of all the approaches at FAR=0.1 and 0.01 for the ROC curves (IJB-A 1:1 verification). The Rank-1, Rank-5, and Rank-10 retrieval accuracies of the CMC curves and TPIR at FPIR = 0.01 and 0.1 (IJB-A 1:N identfication). We also show the results before and after media averaging where m means media averaging.

Besides using the average feature representation, we also perform media averaging which is to first average the features coming the same media (image or video) and then further average the media average features to generate the final feature representation. We

CS2-Verif	DCNN (setup 1)	DCNN (setup 2)	DCNN (setup 3)	DCNN$_m$ (setup 1)	DCNN$_m$ (setup 2)	DCNN$_m$ (setup 3)
FAR=1e-2	0.913 ± 0.008	0.91 ± 0.008	0.922 ± 0.007	0.92 ± 0.01	0.922 ± 0.008	**0.935 ± 0.007**
FAR=1e-1	0.98 ± 0.004	0.967 ± 0.003	0.984 ± 0.003	0.981 ± 0.003	0.968 ± 0.003	**0.986 ± 0.002**
CS2-Ident	DCNN (setup 1)	DCNN (setup 2)	DCNN (setup 3)	DCNN$_m$ (setup 1)	DCNN$_m$ (setup 2)	DCNN$_m$ (setup 3)
Rank-1	0.9 ± 0.01	0.896 ± 0.008	0.909 ± 0.008	0.905 ± 0.007	0.915 ± 0.007	**0.931 ± 0.007**
Rank-5	0.963 ± 0.006	0.954 ± 0.006	0.969 ± 0.006	0.965 ± 0.004	0.959 ± 0.005	**0.976 ± 0.004**
Rank-10	0.977 ± 0.006	0.965 ± 0.004	0.981 ± 0.003	0.977 ± 0.004	0.967 ± 0.004	**0.985 ± 0.002**

Table 5.5: Results on the JANUS CS2 dataset. The TAR of all the approaches at FAR=0.1 and 0.01 for the ROC curves. The Rank-1, Rank-5, and Rank-10 retrieval accuracies of the CMC curves. We report average and standard deviation of the 10 splits. We also show the results before and after media averaging where m means media averaging.

IJB-A-Verif	[146]	JanusB [119]	JanusD [119]	DCNN$_{bl}$ [147]	NAN [148]	DCNN$_{3d}$ [149]
FAR=1e-3	0.514 ± 0.006	0.65	0.49	-	0.785 ± 0.028	0.725
FAR=1e-2	0.732 ± 0.033	0.826	0.71	-	0.897 ± 0.01	0.886
FAR=1e-1	0.895 ± 0.013	0.932	0.89	-	0.959 ± 0.005	-
IJB-A-Ident	[146]	JanusB [119]	JanusD [119]	DCNN$_{bl}$ [147]	NAN [148]	DCNN$_{3d}$ [149]
Rank-1	0.820 ± 0.024	0.87	0.88	0.895 ± 0.011	-	0.906
Rank-5	0.929 ± 0.013	-	-	0.963 ± 0.005	-	0.962
Rank-10	-	0.95	0.97	-	-	0.977
IJB-A-Verif	DCNN$_{pose}$ [150]	DCNN$_m$ (setup 1)	DCNN$_m$ (setup 2)	DCNN$_m$ (setup 3)	DCNNtpe [151]	TP [152]
FAR=1e-3	-	0.704 ± 0.037	0.762 ± 0.038	0.76 ± 0.038	**0.813 ± 0.02**	-
FAR=1e-2	0.787	0.863 ± 0.02	0.885 ± 0.014	0.889 ± 0.016	0.9 ± 0.01	**0.939 ± 0.013**
FAR=1e-1	0.911	0.966 ± 0.05	0.954 ± 0.003	**0.968 ± 0.005**	0.964 ± 0.01	-
IJB-A-Ident	DCNN$_{pose}$ [150]	DCNN$_m$ (setup 1)	DCNN$_m$ (setup 2)	DCNN$_m$ (setup 3)	DCNNtpe [151]	TP [152]
Rank-1	0.846	0.916 ± 0.009	0.923 ± 0.01	**0.942 ± 0.008**	0.932 ± 0.001	0.928 ± 0.01
Rank-5	0.927	0.971 ± 0.007	0.961 ± 0.006	**0.98 ± 0.005**	-	-
Rank-10	0.947	0.981 ± 0.005	0.969 ± 0.004	**0.988 ± 0.003**	0.977 ± 0.005	0.986 ± 0.003

Table 5.6: Results on the IJB-A dataset. The TAR of all the approaches at FAR=0.1, 0.01, and 0.001 for the ROC curves (IJB-A 1:1 verification). The Rank-1, Rank-5, and Rank-10 retrieval accuracies of the CMC curves (IJB-A 1:N identfication). We report average and standard deviation of the 10 splits. All the performance results reported in [119], Janus B (JanusB-092015), Janus D (JanusD-071715), DCNN$_{bl}$ [147], DCNN$_{3d}$ [149], NAN [148], DCNN$_{pose}$ [150], DCNN$_{tpe}$ [151], and TP [152]. The systems have produced results for setup 1 (based on landmarks provided along with the dataset) only. In addition, we also compare the performance of the recent work, DCNN$_{tpe}$ [151] where the performance difference mainly comes from the better preprocessing module and improved metric, [25].

show the results before and after media averaging for both IJB-A and JANUS CS2 dataset in Table 5.4 and in Table 5.5. It is clear that media averaging significantly improves the performance.

Tables 5.6 and 5.7 summarize the scores (*i.e.*, both ROC and CMC numbers) produced by different face verification methods on the IJB-A and JANUS CS2 datasets, respectively. For the IJB-A dataset, we compare our fusion results (*i.e.*, we perform fine-tuning and TSE in Setup 3.) with DCNN$_{bl}$ (bilinear CNN [147]), DCNN$_{pose}$ (multi-pose DCNN models [150]), [148], DCNN$_{3d}$ [149], template adaptation (TP) [152], DCNN$_{tpe}$ [151] and the ones [119] reported recently by NIST where JanusB-092015 achieved the best verification results, and JanusD-071715 the best identification results. For the JANUS CS2 dataset, Table 5.7 includes, a DCNN-based method [146], Fisher vector-based method [57], DCNN$_{pose}$ [150], DCNN$_{3d}$ [149], and two commercial off-the-shelf matchers, COTS and GOTS [128]. From the ROC and CMC scores, we see that the fusion of DCNN methods significantly improve the performance. This can be attributed to the fact that the DCNN model does capture face variations over a large dataset and generalizes well to a new small dataset.

In addition, the performance results of Janus B (Jan-usB-092015), Janus D (JanusD-071715), DCNN$_{bl}$ and DCNN$_{pose}$ systems have produced results for setup 1 (based on landmarks provided along with the dataset) only.

CS2-Verif	COTS	GOTS	FV [57]	DCNN$_{pose}$ [150]
FAR=1e-3	-	-	-	-
FAR=1e-2	0.581±0.054	0.467±0.066	0.411±0.081	0.897
FAR=1e-1	0.767±0.015	0.675±0.015	0.704±0.028	0.959
CS2-Ident	COTS	GOTS	FV [57]	DCNN$_{pose}$ [150]
Rank-1	0.551 ± 0.003	0.413 ± 0.022	0.381 ± 0.018	0.865
Rank-5	0.694 ± 0.017	0.571 ± 0.017	0.559 ± 0.021	0.934
Rank-10	0.741 ± 0.017	0.624 ± 0.018	0.637 ± 0.025	0.949
CS2-Verif	DCNN$_{3d}$ [149]	DCNN (setup 1)	DCNN (setup 2)	DCNN (setup 3)
FAR=1e-3	0.824	0.81 ± 0.018	0.823 ± 0.013	**0.83 ± 0.014**
FAR=1e-2	0.926	0.92 ± 0.01	0.922 ± 0.008	**0.935 ± 0.007**
FAR=1e-1	-	0.981 ± 0.003	0.968 ± 0.003	**0.986 ± 0.002**
CS2-Ident	DCNN$_{3d}$ [149]	DCNN (setup 1)	DCNN (setup 2)	DCNN (setup 3)
Rank-1	0.898	0.905 ± 0.007	0.915 ± 0.007	**0.931 ± 0.007**
Rank-5	0.956	0.965 ± 0.004	0.959 ± 0.005	**0.976 ± 0.004**
Rank-10	0.969	0.977 ± 0.004	0.967 ± 0.004	**0.985 ± 0.002**

Table 5.7: Results on the JANUS CS2 dataset. The TAR of all the approaches at FAR=0.1, 0.01, and 0.001 for the ROC curves. The Rank-1, Rank-5, and Rank-10 retrieval accuracies of the CMC curves. We report average and standard deviation of the 10 splits. The performance results of DCNN$_{pose}$ have produced results for setup 1 only.

5.3.5 Labeled Face in the Wild

We also evaluate our approach on the well-known LFW dataset [105] using the standard protocol which defines 3,000 positive pairs and 3,000 negative pairs in total and further splits them into 10 disjoint subsets for cross validation. Each subset contains 300 positive and 300 negative pairs. It contains 7,701 images of 4,281 subjects. We compare the mean accuracy of the proposed deep model with other state-of-the-art deep learning-based methods: DeepFace [60], DeepID2 [62], DeepID3 [153], FaceNet [11], Yi *et al.* [13], Wang *et al.* [146], Ding *et al.* [154], Parkhi *et al.* [63], and human performance on the "funneled" LFW images. The results are summarized in Table 5.8. It can be seen that our approach performs comparable to other deep learning-based methods. Note that some of the deep learning-based methods compared in Table 5.8 use millions of data samples for training the model. In comparison, we use only the CASIA dataset for training our model which has less than 500K images.

Method	#Net	Training Set	Metric	Mean Accuracy $\pm$ Std
DeepFace [60]	1	4.4 million images of 4,030 subjects, private	cosine	95.92% $\pm$ 0.29%
DeepFace	7	4.4 million images of 4,030 subjects, private	unrestricted, SVM	97.35% $\pm$ 0.25%
DeepID2 [62]	1	202,595 images of 10,117 subjects, private	unrestricted, Joint-Bayes	95.43%
DeepID2	25	202,595 images of 10,117 subjects, private	unrestricted, Joint-Bayes	99.15% $\pm$ 0.15%
DeepID3 [153]	50	202,595 images of 10,117 subjects, private	unrestricted, Joint-Bayes	99.53% $\pm$ 0.10%
FaceNet [11]	1	260 million images of 8 million subjects, private	L2	99.63% $\pm$ 0.09%
Yi *et al.* [13]	1	494,414 images of 10,575 subjects, public	cosine	96.13% $\pm$ 0.30%
Yi *et al.*	1	494,414 images of 10,575 subjects, public	unrestricted, Joint-Bayes	97.73% $\pm$ 0.31%
Wang *et al.* [146]	1	494,414 images of 10,575 subjects, public	cosine	96.95% $\pm$ 1.02%
Wang *et al.*	7	494,414 images of 10,575 subjects, public	cosine	97.52% $\pm$ 0.76%
Wang *et al.*	1	494,414 images of 10,575 subjects, public	unrestricted, Joint-Bayes	97.45% $\pm$ 0.99%
Wang *et al.*	7	494,414 images of 10,575 subjects, public	unrestricted, Joint-Bayes	98.23% $\pm$ 0.68%
Ding *et al.* [154]	8	471,592 images of 9,000 subjects, public	unrestricted, Joint-Bayes	99.02% $\pm$ 0.19%
Parkhi *et al.* [63]	1	2.6 million images of 2,622 subjects, public	unrestricted, TDE	98.95 %
Human, funneled [146]	N/A	N/A	N/A	99.20%
Our $DCNN_S$ + $DCNN_L$	2	490,356 images of 10,548 subjects, public	cosine	98% $\pm$ 0.5%
Our $DCNN_S$ + $DCNN_L$	2	490,356 images of 10,548 subjects, public	unrestricted, TSE	98.33% $\pm$ 0.7%

Table 5.8: Accuracy of different methods on the LFW dataset.

5.3.6 Comparison with Methods based on Annotated Metadata

Most systems compared in this chapter produced the results for setup 1 which is based on landmarks provided along with the dataset only (*i.e.,* except DCNN$_{tpe}$.). For DCNN$_{3d}$ [149], the number of face images is augmented along with the original CASIA-WebFace dataset by around 2 million using 3D morphable models. On the other hand, NAN [148] and TP [152] used datasets with more than 2 million face images to train the model. However, the networks used in this work were trained with the original CASIA-WebFace which contains around 500K images. In addition, TP adapted the one-shot similarity framework [155] with linear support vector machine for set-based face verification and trained the metric on-the-fly with the help of a pre-selected negative set during testing. Although TP achieved significantly better results than other approaches, it takes more time during testing than the proposed method since our metric is trained off-line and requires much less time for testing than TP. We expect the performance of the proposed approach can also be improved by using the one-shot similarity framework. As shown in Table 5.6, the proposed approach achieves comparable results to other methods and strikes a balance between testing time and performance. In a recent work, DCNN$_{tpe}$ [151], adopted a probabilistic embedding for similarity computation and a new face preprocessing module, hyperface [25], for improved face detection and fiducials where [25] is a multi-task deep network trained for the tasks of gender classification, fiducial detection, pose estimation and face detection. We plan to incorporate hyperface into the current framework which may yield some improvement in performance.

5.3.7 Run Time

The DCNN_S model for face verification is trained on the CASIA-Webface dataset from scratch for about 4 days and for DCNN_L, it takes 20 hours to train on the same face dataset which is initialized using the weights of Alexnet pretrained on the ImageNet dataset. The two networks are trained using NVidia Titan X with cudnn v4. The running time for face detection is around 0.7 second per image. The facial landmark detection and feature extraction steps take about 1 second and 0.006 second per face, respectively. The face association module for a video takes around 5 fps on average.

5.4 Open Issues

Given sufficient number of annotated data and GPUs, DCNNs have been shown to yield impressive performance improvements. Still many issues remain to be addressed to make the DCNN-based recognition systems robust and practical. We discussed design considerations for each component of a full face verification system, including

- **Face detection:** In contrast to generic object detection task, face detection is more challenging due to the wide range of variations in the appearance of faces. The variability is caused mainly by changes in illumination, facial expression, viewpoints, occlusions, etc. Other factors such as blurry images and low resolution are prominent in face detection task.

- **Fiducial detection:** Most of the datasets only contain few thousands images. A large scale annotated and unconstrained dataset will make the face alignment sys-

tem more robust to the challenges, including extreme pose, low illumination, small and blurry face images. Researchers have hypothesized that dee-per layers of convnets can encode more abstract information such as identity, pose, and attributes; However, it has not yet been thoroughly studied which layers exactly correspond to local features for fiducial detection.

- **Face association:** Since the video clips may contain media of low-quality images, the blurred and low-resolution image makes the face detection not reliable. This may lead to performance degradation of face association since a face track will not be initiated due to the missing of face detection. Besides, abrupt motion, occlusion, and crowded scene can lead to performance degradation of tracking and potential identity switching.

- **Face verification:** For face verification, the performance can be improved by learning a discriminative distance measure. However, due to memory constraints limited by graphics cards, how to choose informative pairs or triplets and train the network end-to-end using online methods (*e.g.*, stochastic gradient descent) on large-scale datasets is still an open problem.

5.5 Summary

We presented the design and performance of our automatic face verification system, which automatically locates faces and performs verification/recognition on newly released challenging face verification datasets, IARPA Benchmark A (IJB-A) and its extended version, JANUS CS2. It was shown that the proposed DCNN-based system can not

only accurately locate the faces across images and videos but also learn a robust model for face verification. Experimental results demonstrate that the performance of the proposed system on the IJB-A dataset is much better than a FV-based method and some COTS and GOTS matchers.

Chapter 6: A Cascaded Convolutional Neural Network for Age Estimation of Unconstrained Faces

6.1 Overview

Besides face recognition, we would like to utilize the trained DCNN model for other face-related analysis, and we focus on apparent age estimation in this chapter. Traditionally, the problem is tackled through pure classification or regression approaches. In this chapter, we present a cascaded approach which incorporates the advantages of both classification and regression approaches. Given an input image, we first apply the age group classification algorithm to obtain a rough estimate and then perform age group specific regression to obtain an accurate age estimate.

Like other facial analysis techniques, age estimation is affected by many intrinsic and extrinsic challenges, such as illumination variation, race, attributes, etc. One may define the age estimation task as a process of automatically labeling face images with the exact age, or the age group (age range) for each individual. It was suggested in [156] to differentiate the problem of age estimation along four concepts:

- Actual age: real age of an individual.
- Appearance age: age information shown on the visual appearance.

Figure 6.1: Estimated age on sample images from [2]. Our method is able to predict the age in unconstrained images with variations in pose, illumination, age groups, and expressions.

- Apparent age: suggested age by human subjects from the visual appearance.
- Estimated age: recognized age by an algorithm from the visual appearance.

The proposed cascaded classification and regression approach for apparent age estimation is based on a deep convolutional neural network. Our method consists of three main stages: (1) a single coarse age classifier, (2) multiple age regressors, and (3) an error correcting stage to correct the mistakes made by the age group classifer. Since the number of samples for apparent age estimation is limited, we exploit a DCNN model pretrained for large-scale face identification task and finetune the model for age group classification and age regression tasks. This strategy is effective since the face recognition model trained on the CASIA-WebFace dataset [13] (*i.e.* it consists of 10,575 subjects and 494,414 images.) encodes rich information reflecting large variations in facial appearances due to aging and variations in pose, expression and illumination.

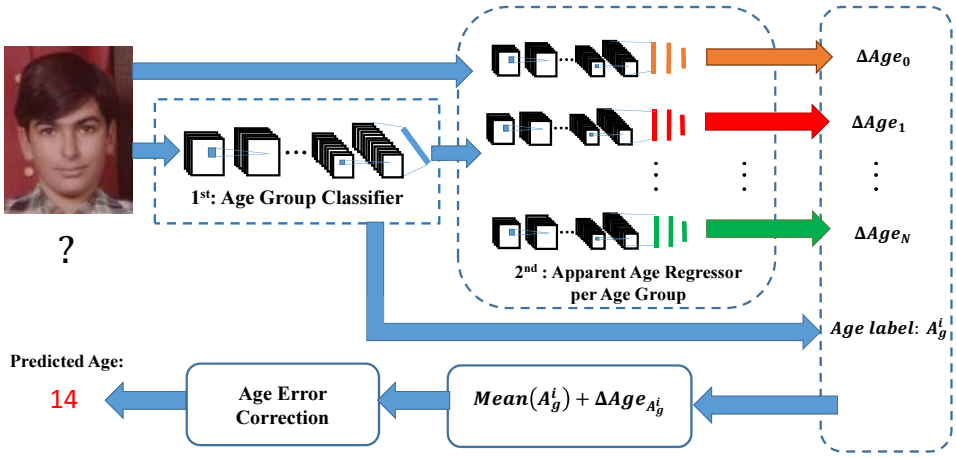


Figure 6.2: An overview of the proposed age cascade apparent age estimator.

The contribution of this chapter is to propose the age error correction module which mitigates the common disadvantage of coarse-to-fine approaches. Typically, the errors made at the initial classification stage cannot be recovered by the regressors at the following stage. In this work, we set up the baseline algorithm which is based on the proposed regression algorithm in Section 6.2.6 and study how the coarse-to-fine strategy and the error correction module improve the prediction performance. Figure 6.2 presents an overview of the proposed age estimation method.

The rest of the chapter is organized as follows: The proposed approach is presented in Section 6.2 with a concrete example. Experimental results are provided in Section 6.3, and Section 6.4 concludes the chapter with a brief summary and discussion.

6.2 Proposed Method

Figure 6.2 shows an overview of our CNN-based cascaded age estimation method. Our approach consists of three main components: (1) age group classifier, (2) age re-

gressor to predict the relative age with respect to each age group mean, and (3) apparent age error correction. Given a face image, we first apply the age group classifier to get a rough estimate of the age range from the image. Then, we choose the corresponding age regressor based on the classification results to predict the relative age with respect to the predicted group mean and combine them to get the apparent age estimate. Then, we utilize the characteristic of the classification plus regression framework to design an age error correction scheme to correct age classification and regression errors. Finally, the algorithm outputs the final age estimate for the given input image. In what follows next, we will describe each of these component in detail.

6.2.1 Face Preprocessing

In our work, all the face detection and facial landmark detection are handled using the open source library dlib [14] [157]. Three landmark points (the center of the left eye, the center of the right eye, and the nose base) are used to align the detected faces into the canonical coordinate system using the similarity transform.

6.2.2 Deep Face Feature Representation

We use the DCNN model with the architecture similar to the one proposed in [13] which is pretrained for the face-identification task with softmax loss using the CASIA-WebFace dataset [13]. The CASIA-WebFace dataset consists of 10,575 subjects and 494,414 images. The architecture is composed of 10 convolutional layers, 5 pooling layers and 1 fully connected layer. In our work, we use PReLU [123] instead of ReLU

as the nonlinear activation function and data augmentation to train the network. The input is a color image of aligned faces of dimension $100 \times 100 \times 3$. The details of this architecture are given in Table 6.1. We do net surgery on this network (*i.e.*, we cut off the part after pool5 layer.) and use its pretrained weights on the CASIA-WebFace dataset to finetune on the age group dataset and apparent age estimation dataset to perform age group classification and relative age regression with respect to each age group.

6.2.3 Age Group Classifier

Inspired by the Viola and Jones face detection algorithm [14], we quantize the human age into several age groups (*e.g.* 0-7, 8-14, 15-23, etc.) which is an easier problem than directly performing classification or regression for the whole age range which requires a large amount of training data. To train the age group classifier, we remove the original fully connected layer, add the PReLU units and the fully connected layer with 512 outputs and finetune it on the the Images of Groups [158], Adience [159] and FGNet [160] datasets to obtain the DCNN-based age group classifier.

6.2.4 Apparent Age Regressor Per Age Group

To train the age regressor for each age group, we prepare the training data by splitting each training sample into the corresponding age group based on its ground truth age, and then subtract the mean of that group. The regressors are trained in two ways. The first one is to extract the pool5 features and use them to train the regressors with a large batch size. The other is to train the regressor through end-to-end network finetuning but with

a smaller batch size. (*i.e.*, Similarly, we keep the part before pool5 layer and add fully connected layers.) Since the pool5 feature in the face identification task is followed by the fully connected layer with 10,575 output corresponding to the number of subject in the CASIA-WebFace dataset, the pool5 features should contain strong discriminative information from all the face images to classify a large number of subjects in the training data. In addition, we also adopt a novel loss function called, the Gaussian Loss, which takes the a rough age (*i.e.* the age is represented as a mean and a standard derivation instead of the exact age) as input and is robust for apparent age estimation. The role of the new loss function in learning the nonlinear regression method is discussed in Section 6.2.6.

For the pre-training of DCNN face representation model, we use the standard batch size 128 for the training phase. The initial negative slope for PReLU is set to 0.25 as suggested in [123]. The weight decay rates of all the convolutional layers are set to 0, and the weight decay of the final fully connected layer to 5e-4. In addition, the learning rate is set to 1e-2 initially and reduced by half every 100,000 iterations. The momentum is set to 0.9. Finally, we use the snapshot of 1,000,000th iteration as our pretrained model. For the finetuning of the age group classifier, we use the learning rate, 1e-4, for the convolutional layers and 1e-3 for the fully connected layers with 100,000 iterations. For training each age regressor, we first extract all the 320-d feature vectors for each age group and feed them at once into the age regressor network. We train it with 30,000 iterations using the learning rate, 1e-2, and momentum, 0.9. For the end-to-end finetuning of the regressors, we use batch size, 128, with the learning rate, 1e-4, for the convolutional layers and 1e-3 for the fully connected layers. The 120,000th models are used for each age regressor. Data augmentation is performed by randomly cropping 100×100 regions from a $128 \times$

128 box and horizontally face flipping.

6.2.5 Age Error Correction

In practice, the age group classifier will make errors and these errors significantly affect the final age estimation results for the second stage regressors. To handle these errors, we employ an error correcting approach. When we train the regressor for each age group, we also include the training examples from the neighboring age group. For example, given 3 age groups, (1) 8-14, (2) 15-21, and (3) 22-28, if we want to train the age regressor for the first age group, besides the training samples with ages ranging from 8 to 14 years old, we also add the training samples from its neighboring group (*i.e.*, we added the samples from ± 2 groups for the experiments.), that is the second age group. Thus, when the classifier mistakenly assigns the subject to the neighboring age group, the regressor is able to predict a large enough value and correct the error caused by the age group classifier. Furthermore, to take the classifier error into consideration, we also add the misclassified samples to augment the training samples of all the regressors in between the true and wrong groups to increase the chance of correcting the imprecise age estimate so that it is close to the ground truth through our error correction scheme. The detailed step-by-step illustration for the age error correction scheme and other components will be presented in the following subsection. The pseudo code for our age correction approach is given in Algorithm 3.

Name	Type	Filter Size/Stride	#Params
Conv11	convolution	3×3×1 / 1	0.84K
Conv12	convolution	3×3×32 / 1	18K
Pool1	max pooling	2×2 / 2	
Conv21	convolution	3×3×64 / 1	36K
Conv22	convolution	3×3×64 / 1	72K
Pool2	max pooling	2×2 / 2	
Conv31	convolution	3×3×128 / 1	108K
Conv32	convolution	3×3×96 / 1	162K
Pool3	max pooling	2×2 / 2	
Conv41	convolution	3×3×192 / 1	216K
Conv42	convolution	3×3×128 / 1	288K
Pool4	max pooling	2×2 / 2	
Conv51	convolution	3×3×256 / 1	360K
Conv52	convolution	3×3×160 / 1	450K
Pool5	avg pooling	7×7 / 1	
Dropout	dropout (40%)		
Fc6	fully connection	10575	3305K
Cost	softmax		
total			5015K

Table 6.1: The base architecture of DCNN model used in this chapter [13] to finetune on the age group classification and Δage regression for each age group.

Algorithm 3 AGE ESTIAMTION ALGORITHM

Input: (a) Input face image, I, (b) maxIter iterations, (c) age group classifier, G_0, and age regressor per age group, A_0, A_1, ..., A_{N-1} where N is the number of age groups and both age group classifier and age regressors are all DCNN-based models.

Output: Predicted apparent age, $\hat{a}$.

$g_\ell = G_0(I)$, where g_ℓ is the predicted age group label.
For i = 0 to N-1
    $\Delta a_i = A_i(I)$.
End For
$\hat{a}$ = mean(g_ℓ)+Δa_{g_ℓ}.
// Age estimation error correction
For i = 0 to maxIter - 1
    $\hat{g}_\ell = L(\hat{a})$, where $L(\cdot)$ returns the age group label of $\hat{a}$.
    IF $\hat{g}_\ell = g_\ell$
        Return $\hat{a}$
    ELSE
        $\hat{a}$ = mean($\hat{g}_\ell$)+$\Delta a_{\hat{g}_\ell}$
    End IF
    $g_\ell = \hat{g}_\ell$
End For
Return $\hat{a}$

6.2.6 Non-linear Regression

We use a 3-layer neural network to learn the age regressor for each age group. The number of layers is determined experimentally to be 3. The regression is learned by optimizing the Gaussian loss function as follows [2]. The Gaussian loss function is useful since the apparent age labels are usually not exact.

$$L = \frac{1}{N}\sum_{i=1}^{i=N} 1 - e^{-\frac{(\Delta x_i - \mu_i)^2}{2\sigma_i^2}}, \tag{6.1}$$

where L is the average loss for all the training samples, Δx_i is the predicted shift in age from the mean of the corresponding age group. μ_i is the ground truth shift in age and σ_i is the standard deviation in age increment for the i^{th} training sample. The network

parameters are trained using the back-propagation algorithm [161] with batch gradient descent. The gradient obtained for the loss function is given by (6.2). This gradient is used for updating the network weights during training using back-propagation.

$$\frac{\partial L}{\partial \Delta x_i} = \frac{1}{N\sigma^2}(\Delta x_i - \mu_i) e^{-\frac{(\Delta x_i - \mu_i)^2}{2\sigma_i^2}}. \tag{6.2}$$

We apply dropout [162] after each fully connected layers to reduce the over-fitting due to the limited number of training data. The amount of dropout applied is 0.4, 0.3 and 0.2 for the input, first and second layers of the network respectively. The dropout ratio is applied in a decreasing manner to cope up with the decrease in the number of parameters for the deeper layers. Each layer is followed by the (PReLU) [123] activation function except the last one which predicts the age. The first layer is the input layer which takes the 320 dimensional feature vector obtained from the face-identification task. The output of this layer, after the dropout and PReLU operation, is fed to the fist hidden layer containing 320 hidden units. Subsequently, the output propagates to the second hidden layer containing 160 hidden units. The output from this layer is used to generate a scalar value that would describe the apparent age. Figure 6.3 depicts the 3-layer neural network used.

6.2.7 A Toy Example

To illustrate the end-to-end pipeline of the proposed age estimation algorithm, we present a toy example below. In this example, we use the 3 age group setting for the age group classifier where (1) the first age group is from 8 to 14 years, (2) the second 15 to 21, and (3) the third 22 to 28. The age regressor will predict Δage with respect to the mean

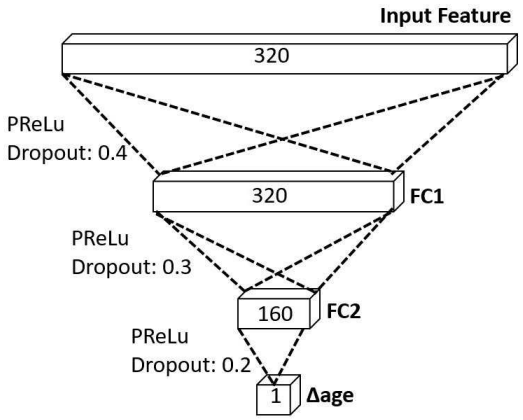


Figure 6.3: The 3-layer neural network used for estimating the increment in age for each age group.

age of its corresponding group. For example, the regressor for the first age group takes charge of predicting the real value ranging from -3 (*i.e.* 8 - 11 = -3, where 11 is the mean age of the first group) to +3 (*i.e.* 14 - 11 = 3). Now, given a face image with ground truth age 27 years old, ideally the predicted age group label should be 3 after passing the image into the age group classifier. Then, we will use the third age regressor to predict its Δage which should ideally predict the value as +2 and then we can estimate the apparent age as 25 + 2 = 27 by combining the results of the age group classifier and its corresponding age regressor where 25 is the group mean for the third age group. However, as mentioned in Section 6.2.5, in practice, if the age group classifier makes mistakes, the age estimation results will be wrong. To handle this error, we do the age error correction as described in Section 6.2.5. Now, given another face image with ground truth age 14, incorrectly being classified into third age group, we augment the misclassified samples when we train the regressor. Thus, it can be expected that the Δage should be negative enough, say -5, and as a result, the age estimation will be 25 - 5 = 20 which is still wrong but falls in the range of the second group. Then, we can pass the image again to the second group regressor to

get a new estimate, say 18 - 4 = 14. We stop correcting the error when the predicted age and the previous predicted age falls in the same group or reach the maximum number of iterations. That is, we will pass the image to the first regressor again and it will predict 11 + 3 = 14 and then we stop. Otherwise, we continue to perform the correction.

The proposed age estimation algorithm is summarized in Algorithm 3. The execution orders for both the classification and regression parts are written in parallel, and thus it runs in one age group classification plus N Δage regression simultaneously in total. The maximum number of iterations is preset to avoid looping.

6.3 Experimental Results

We evaluate the proposed method on two publicly available datasets: Adience [159] and FG-Net [160]. Both datasets include unconstrained images of individuals which are labeled by their actual biological ages. In addition to these two datasets, we present results on the ICCV 2015 Chalearn 'Looking at people-Age Estimation' challenge dataset [2]. The main difference between this dataset and Adience and FG-Net datasets is that Chalearn includes unconstrained images of individuals labeled by their apparent ages.

6.3.1 Datasets

Adience dataset [159] consists of $26,580$ unconstrained images of $2,284$ subjects in 8 age groups (0-2, 4-6, 8-13, 15-20, 25-32, 38-43, 48-53, 60+). The standard five-fold, subject-exclusive cross-validation protocol is used for testing (*i.e.*, we merge 0-2 and 4-6 into one for the experiments of Challenge and FG-Net datasets.)

FG-Net aging dataset [160] contains a collection of $1,002$ images of 82 subjects, where each image is annotated with true age.

Images of groups [158] consists of $28,231$ faces in $5,080$ images. Each face is annotated with a label corresponding to one of the seven age groups; 0-2, 3-7, 8-12, 13-19, 20-36, 37-65, 66+ .

Chalearn Workshop Challenge dataset is the first dataset on apparent age estimation containing annotations. The dataset consists of $2,476$ training images, $1,136$ validation images, and $1,087$ test images, which were taken from individuals aged between 0 to 100. The images are captured in the wild, with variations in pose, illumination and quality. Figure 6.4 shows the distribution of the 'Chalearn Looking at People' Challenge dataset across the different age groups. It is evident from this figure that most of the data are distributed around the age group of 20-50, while there are very few samples in the range of 0-15 and above 55. The remaining data consists of the test set which has not been released publicly.

6.3.2 Experimental Details

For the first stage of age classification, we augmented the training set with the training splits of Adience [159], FG-Net [160] and Images of groups [158] datasets. To evaluate on the FG-Net, we train the seven regressor networks and then pass them through our proposed error correcting mechanism to predict the final age. Although the recently released IMDB-WIKI dataset [163] contains a large collection of images with ages, the number of the images for the young and old age groups is much smaller than other groups

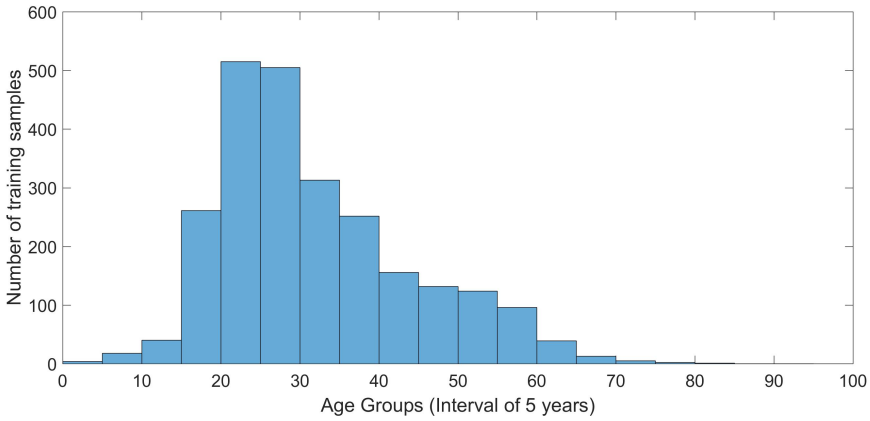


Figure 6.4: Training data distribution of ICCV-2015 Chalearn Looking at People Apparent Age Estimation Challenge, with regard to age groups.

and some of the annotations for the dataset are noisy. Due to these factors, we confine the age group ranges to the ones defined by Adience [159] and focus on those previosly well-labelled datasets for the proposed approach. The study of the influences by different ranges of age group intervals is left for future work. All the models were trained using Caffe [164]. We also compare the performance of our proposed method with a recently proposed geometry-based method [86], which is referred to as Grassmann-Regression (G-LR).

6.3.3 Results

To evaluate the performance of age classification algorithm, we conduct experiments on the Adience dataset [159], by following the 5 fold cross validation protocol described in [165]. From Table 6.2, it can be seen that our approach achieve better performance than the previous state-of-the-art methods. In addition, we also visualize what the neurons of DCNN model actually learn after fine-tuning on facial age group dataset

using deepDraw [3]. From the figure 6.5(a) and (e), we can clearly see the appearance and shape of children and the elder. This demonstrates that the DCNN model does adapt the representation for age after fine-tuning. One thing worth noticing is that the accuracy for exact age group classification is around 53%, but the 1-off accuracy is 88.45% (*i.e.*, 1-off means the predicted label is within the neighboring groups of the true one, and 2-off means $\pm$ 2 groups). The results demonstrate the need of our error correction module to make the coarse-to-fine strategy to work better.

Method	**Exact**	**1-off**
Best from [159]	45.1 ± 2.6	79.5 ± 1.4
Best from [165]	50.7 ± 5.1	84.7 ± 2.2
Ours	$\mathbf{52.88 \pm 6}$	$\mathbf{88.45 \pm 2.2}$

Table 6.2: Age estimation results on the Adience benchmark. Listed are the mean accuracy $\pm$ standard error over all age categories. Best results are marked in bold.

After age group classification, we evaluated the performance of the proposed method following the protocol provided by the Chalearn 'Looking at People' challenge dataset to further investigate how the coarse-to-fine strategy and error correction mechanism help the age estimation. The error is computed as follows:

$$\varepsilon = 1 - e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{6.3}$$

where x is the estimated age, μ is the provided apparent age label for a given face image, average of at least 10 different user opinions, and σ is the standard deviation of all (at least 10) gauged ages for the given image. We evaluate our method on the validation set of the challenge [2], as the test set annotations are not available for performing analysis. Our

(a) (b) (c) (d)

(e) (f) (g)

Figure 6.5: We visualize the results for the fine-tuned DCNN model on age group classification using deepDraw [3]. (a) age from 0 to 6 years old, (b) 8 to 13, (c) 15 to 20, (d) 25 to 32, 38 to 43, (e) 48 to 53, and (e) 60+. From the figures, we can clearly see the shape and appearance of children from (a) and of the elder from (e). It demonstrates that the DCNN model does adapt the representation for age after fine-tuning.

baseline approach is to perform age estimation by a single deep regressor (as described in Section 6.2.6) on top of all the DCNN features. From Table 6.3, it shows that the coarse-to-fine strategy improves the prediction results of the baseline approach, and the error correction module further significantly boosts the performance which also demonstrates that the error correction module effectively fixes the errors made by the age classification step. In addition, we also show that the results of end-to-end finetuning on the training data of the challenge data for both baseline and our approach outperform the ones which are trained separately. (*i.e.*, For the results of baseline with end-to-end finetuning, we use the 500,000th model which are trained with the same batch size and learning rate for the proposed approach.) Some prediction sample results from this dataset are shown in Figure 6.6.

Method	Gaussian Error
G-LR [86]	0.62
Baseline	0.39
Our method without error correction	0.382
Our method with error correction	0.355
Baseline with end-to-end finetuning	0.312
Our method with end-to-end finetuning and error correction	0.297

Table 6.3: Performance comparison on the Chalearn Challenge dataset.

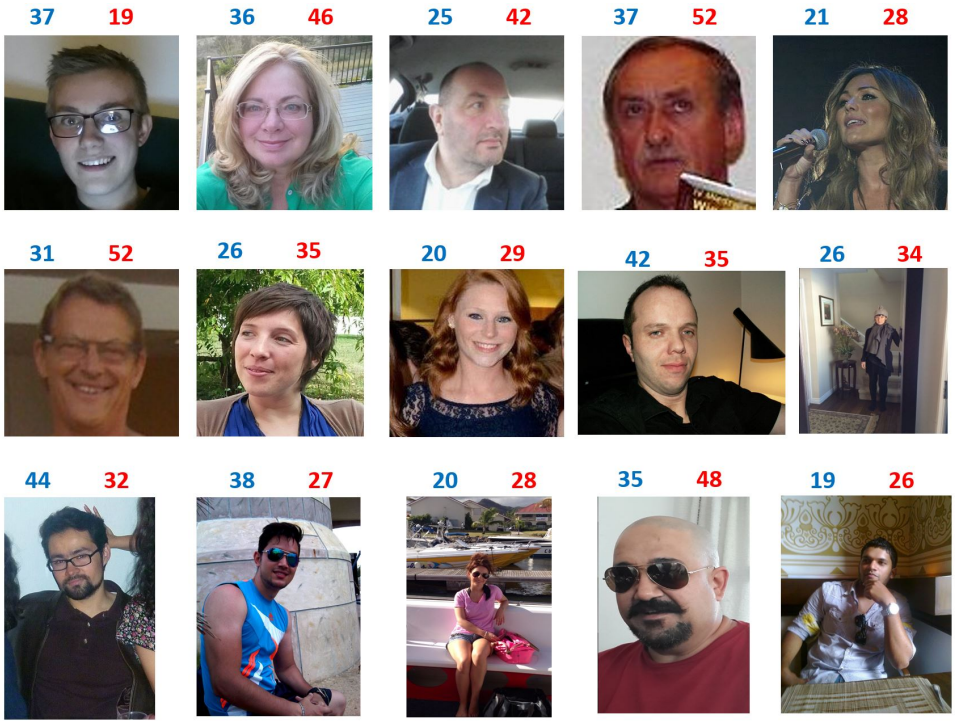


Figure 6.6: Age estimates on the Chalearn Validation set. The incorrect age obtained without using the self correcting module is shown in blue, while the corrected age is given in red.

By looking at the images, we can infer that our method is robust to pose and resolution changes to a certain extent. It fails mostly for extreme illumination and extreme pose scenarios. On further inspection of the Chalearn challenge dataset, we observe the the first stage classification fails to classify correctly when the images have attributes such as hats, glasses, microphone, etc. However, the proposed error correcting mechanism makes it robust to such artifacts. The performance of our method can be improved considerably if we train using large-scale age labeled data.

Finally, we further evaluate the proposed method with end-to-end finetuning on the FG-Net dataset (*i.e.*, For FGNet, we set $\sigma = 2$ for Gaussian loss.). Since the training of DCNN is computationally intensive, a fair amount of time is needed to complete the full leave-one-person out (LOPO) evaluations. Thus, we chose to compromise and show a result that demonstrates the performance level as compared to other methods. We randomly chose 73 subjects and used their images as the training data and the rest for testing. Table 6.4 shows the empirical evaluation of our method compared with several other methods proposed in recent years (*i.e.,* Since the test protocol is different from LOPO used for other methods, the results of the proposed method are not directly comparable to others but only as an empirical performance evaluation.). From this table, it can be seen that our method performs comparable to other state-of-the-art age estimation methods. The approach with error correction module performs much better than the one without considering neighboring samples for error correction during training.

Reference	Method	Training/Testing method	Result (MAE)
Luu2009 [166]	2 stage SVR in AAM subspace	802 training 200 test images	4.37
Ylioinas2013 [167]	LBP Kernel Density Estimate	LOPO	5.09
Geng2013 [168]	Label Distribution (CPNN)	LOPO	4.76
Chen2013 [169]	Cumulative Attribute SVR	LOPO	4.67
El Dib2010 [170]	Enhanced Biologically -Inspired features	LOPO	3.17
Han2013 [160]	Component and holistic BIF	LOPO	4.6
Hong2013 [171]	Biologically InspiredAAM	LOPO	4.18
Chao2013 [172]	Label-sensitive learning	LOPO	4.38
Ours proposed method	**Classification+Regression**	**890 train , 112 test**	**4.8**
Ours proposed method	**Classification+Regression+Error Correction**	**890 train , 112 test**	**3.49**

Table 6.4: Performance comparison of different age estimation algorithms on the FG-Net aging database using mean absolute error(MAE). Since the training of DCNNs is computationally intensive, the evaluation of the proposed approach does not follow the full LOPO protocol. The results are for an empirical evaluation to show the performance level of the proposed approach.

6.3.4 Runtime

All the experiments were performed using NVIDIA GTX TITAN-X GPU and the CuDNN library on a 2.3Ghz computer. The first stage training for the classification task took approximately 8 hours whereas training for the second stage took approximately 8 hours per regressor. The system is fully automated with minimal human intervention. The end-to-end system takes about 2.5 seconds per image for age estimation, with only 0.8 seconds being spent in age estimation given the aligned face while the remaining time being spent on face detection and alignment.

6.4 Summary

For this chapter, we proposed a cascaded classification-regression framework to perform unconstrained facial apparent age estimation. The proposed approach estimates the apparent age in a coarse-to-fine manner. The age group classifier gives the rough age estimate, the regressor per age group gives the fine-grained age estimate, and the age

error correcting module fixes incorrect prediction. Our experimental results demonstrate the effectiveness of the proposed approach, especially when only a limited number of training data available in the target domain.

Although our age classifiers and regressors are all based on DCNN, our framework is generic and can be extended to other non-DCNN models. In addition, the same classification-regression framework can be also applied to other vision problems, such as head pose estimation.

Chapter 7: Conclusion and Directions for Future Work

In this dissertation, we proposed several approaches to learn robust representations for the face recognition task, including (1) dictionary learning and sparse representation, (2) dense local feature aggregation based on Fisher vector, and (3) deep learning based on deep convolutional neural network. We have thoroughly evaluated each approach and developed an automated system for face verification based on deep convolutional neural networks which yield much better performance against large pose, illumination, and other variations than state-of-the-art methods. Furthermore, we also demonstrated that the learned model for face recognition can be adapted to other face-related task without as many annotation data as face recognition (facial age estimation) and can still yield satisfactory performance.

We also outline several possible directions in which the problems addressed in this dissertation can be further explored.

1. **A Real-time End-to-End Face Verification System:** The automated system developed in the dissertation, it is the result of direct combination of different components. However, Liu *et al.* [118] proposes an single-shot object detector (SSD) based fully convolutional neural network in real-time performance (*i.e.* for a 300 $\times$ 300 image, it can reach more than 40fps.) We have already developed a multi-task

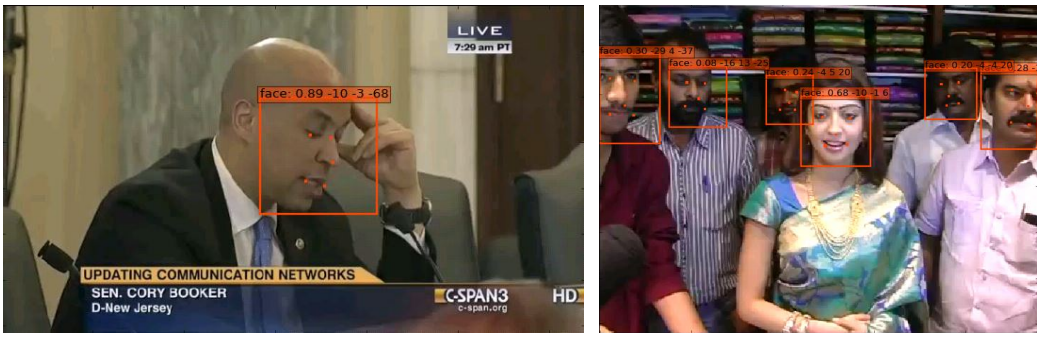
face detector based on SSD which is able to detect five-point fiducial points, face bounding boxes, and head pose. Sample results are shown in Figure 7.1. Furthermore, it is possible to combine it with supervised transformer network [173] which jointly learns fiducial points and the canonical coordinates of the aligned face along with the DCNN model proposed in this dissertation for designing a real-time end-to-end face verification. It not only makes the training of a face verification algorithm easier but also has a practical value for visual surveillance, especially for a Pan-Tilt-Zoom camera network which usually requires real-time vision modules to steer the cameras to the target.

2. **Landmark-based Deep Convolutional Network for Face Verification:** Although the DCNN model achieves promising results for face verification, it is based on a holistic face. In order to effectively handle pose variations, it is useful to incorporate the local feature model (*e.g.*, Fisher vector). Chen and Zheng *et al.* [174] has combined deep convolutional features with Fisher vector for face verification. The other potential direction is to utilize the fiducial points detected by multi-task face detector to develop a deep-fusion network which fuses the deep features around each fiducial points into a pose-robust representation for faces.

3. **Robust Objective Function to Train a DCNN Model on Large-scale Noisy Dataset:** Due to the prevalence of the deep learning, more and more large-scale datasets are available for training the DCNN model for different tasks. (*e.g.*, the MS-Celeb-1M dataset [175] for face recognition contains 99,892 identities from the 1M celebrity list and 8,456,240 images in total. Although there are a lot of face images, there are

also a lot of label errors in the dataset.) Directly training the model on them usually yields lower performance. It is thus interesting and important to develop a robust objective which can not only handle the dataset noise but also learn a meaningful representation at the same time. This could save a lot of time and efforts in cleaning the datasets.

4. **Utilize Motion Information for Video-based Face Related Tasks:** Motion information is not fully explored in this dissertation since we use the average pooling to aggregate the features across frames which may have already loose a lot of motion. However, motion information is definitely important for facial expression analysis. It is interesting to explore the role of motion for face-related applications.



(a) (b)

Figure 7.1: Sample results for our multi-task single shot face detector.

Bibliography

[1] A. Kumar, R. Ranjan, V. Patel, and R. Chellappa. Face alignment by local deep descriptor regression. *arXiv preprint arXiv:1601.07950*, 2016.

[2] S. Escalera, J. Fabian, P. Pardo, X. Baro, J. Gonzalez, H.J. Escalante, and I. Guyon. Chalearn 2015 apparent age and cultural event recognition: datasets and results.

[3] Deepdraw, https://github.com/auduno/deepdraw.

[4] W. Y. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003.

[5] A. Coates, A. Y. Ng, and H. L. Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, pages 215–223, 2011.

[6] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2559–2566, 2010.

[7] J. R. Beveridge, H. Zhang, P. J. Flynn, Y. Y. Lee, V. E. Liong, J. W. Lu, M. de Assis Angeloni, T. de Freitas Pereira, H. X. Li, G. Hua, V. Struc, J. Krizaj, and P. J. Phillips. The ijcb 2014 pasc video face and person recognition competition. In *International Joint Conference on Biometrics*, 2014.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[11] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *arXiv preprint arXiv:1503.03832*, 2015.

[12] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

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

[14] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[15] Y. Wei X. Cao D. Chen, S. Ren and J. Sun. Joint cascade face detection and alignment. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *European Conference on Computer Vision*, volume 8694, pages 109–122. 2014.

[16] J. Li and Y. Zhang. Learning surf cascade for fast and accurate object detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3468–3475, June 2013.

[17] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5325–5334, June 2015.

[18] X. G. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2879–2886. IEEE, 2012.

[19] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *European Conference on Computer Vision*, volume 8692, pages 720–735. 2014.

[20] S. Yang, P. Luo, C. C. Loy, and X. Tang. From facial parts responses to face detection: A deep learning approach. *IEEE International Conference on Computer Vision*, 2015.

[21] S. S. Farfade, M. J. Saberian, and L.-J. Li. Multi-view face detection using deep convolutional neural networks. In *International Conference on Multimedia Retrieval*, 2015.

[22] G. Ross. Fast r-cnn. In *IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[23] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[24] R. Ranjan, V. M. Patel, and R. Chellappa. A deep pyramid deformable part model for face detection. In *IEEE International Conference on Biometrics: Theory, Applications and Systems*, 2015.

[25] R. Ranjan, V. M. Patel, and R. Chellappa. HyperFace: A Deep Multi-task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition, March 2016.

[26] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):681–685, 2001.

[27] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[28] D. Cristinacce and T. F. Cootes. Feature detection and tracking with constrained local models. In *British Machine Vision Conference*, volume 1, page 3, 2006.

[29] A. Asthana, S. Zafeiriou, S. Y. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3444–3451, 2013.

[30] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression, July 3 2014. US Patent App. 13/728,584.

[31] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1078–1085. IEEE, 2010.

[32] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3476–3483, 2013.

[33] Y.-C. Chen, V. M. Patel, P. J. Phillips, and Rama Chellappa. Dictionary-based face recognition from video. In *European Conference on Computer Vision (ECCV)*, 2012.

[34] J. H. Yoon, M.-H. Yang, J. Lim, and K.-J. Yoon. Bayesian multi-object tracking using motion context from multiple objects. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015.

[35] S. Duffner and J. Odobez. Track creation and deletion framework for long-term online multiface tracking. *IEEE Transactions on Image Processing*, 22(1):272–285, Jan. 2013.

[36] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[37] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *European Conference on Computer Vision (ECCV)*, 2008.

[38] M. Roth, M. Bauml, R. Nevatia, and R. Stiefelhagen. Robust multi-pose face tracking by multi-stage tracklet association. In *International Conference on Pattern Recognition (ICPR)*, 2012.

[39] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[40] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[41] F. Comaschi, S. Stuijk, T. Basten, and H. Corporaal. Online multi-face detection and tracking using detector confidence and structured SVMs. In *IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS)*, 2015.

[42] M. Du and R. Chellappa. Face association across unconstrained video frames using conditional random fields. In *European Conference on Computer Vision (ECCV)*, 2012.

[43] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, Sep. 2009.

[44] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994.

[45] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 983–990. IEEE, 2009.

[46] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.

[47] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2012.

[48] P. Wang and Q. Ji. Robust face tracking via collaboration of generic and specific models. *IEEE Transactions on Image Processing*, 17(7):1189–1199, 2008.

[49] Y. M. Lui, J. R.Beveridge, and L. D. Whitley. Adaptive appearance model and condensation algorithm for robust face tracking. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(3):437–448, 2010.

[50] 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.

[51] X. Y. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing*, 19(6):1635–1650, 2010.

[52] L. Wolf, T. Hassner, and Y. Taigman. Effective unconstrained face recognition by combining multiple descriptors and learned background statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1978–1990, 2011.

[53] B. C. Zhang, S. G. Shan, X. L. Chen, and W. Gao. Histogram of Gabor phase patterns (hgpp): a novel object representation approach for face recognition. *IEEE Transactions on Image Processing*, 16(1):57–68, 2007.

[54] 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.

[55] D. Chen, X. D. Cao, F. Wen, and J. Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[56] C. Ding, J. Choi, D. Tao, and L. S. Davis. Multi-directional multi-level dual-cross patterns for robust face recognition. *arXiv preprint arXiv:1401.5311*, 2014.

[57] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher vector faces in the wild. In *British Machine Vision Conference*, volume 1, page 7, 2013.

[58] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman. A compact and discriminative face track descriptor. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[59] J. Lu, V. E. Liong, G. Wang, and P. Moulin. Joint feature learning for face recognition. *IEEE Transactions on Information Forensics and Security*, PP(99):1–1, 2015.

[60] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[61] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems*, pages 1988–1996, 2014.

[62] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. *arXiv preprint arXiv:1412.1265*, 2014.

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

[64] S. H. Zhou, V. Krueger, and R. Chellappa. Probabilistic recognition of human faces from video. *Computer Vision and Image Understanding*, 91(1):214–245, 2003.

[65] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman. Visual tracking and recognition using probabilistic appearance manifolds. *Computer Vision and Image Understanding*, 99(3):303–331, 2005.

[66] X. M. Liu and T. H. Chen. Video-based face recognition using adaptive hidden markov models. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–340, 2003.

[67] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2273–2286, 2011.

[68] H. Cevikalp and B. Triggs. Face recognition based on image sets. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2567–2573, 2010.

[69] R. P. Wang, H. M. Guo, L. S. Davis, and Q. H. Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2496–2503, 2012.

[70] Y. Q. Hu, A. S. Mian, and R. Owens. Sparse approximated nearest points for image set classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 121–128, 2011.

[71] Y.-C. Chen, V. M. Patel, P. J. Phillips, and R. Chellappa. Dictionary-based face recognition from video. In *European Conference on Computer Vision*, pages 766–779. 2012.

[72] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[73] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *IEEE International Conference on Computer Vision*, pages 498–505, 2009.

[74] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2005.

[75] Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *British Machine Vision Conference*, pages 1–12, 2009.

[76] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *International Conference on Machine learning*, pages 209–216, 2007.

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

[78] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *IEEE International Conference on Computer Vision*, pages 365–372, 2009.

[79] D. Chen, X. D. Cao, L. W. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *European Conference on Computer Vision*, pages 566–579. 2012.

[80] J. Hu, J. Lu, and Y.-P. Tan. Discriminative deep metric learning for face verification in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1875–1882, 2014.

[81] Z. Huang, R. Wang, S. Shan, and X. Chen. Projection metric learning on Grassmann manifold with application to video based face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 140–149, 2015.

[82] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[83] S. Ramanathan, B. Narayanan, and R. Chellappa. Computational methods for modeling facial aging: A survey. *Journal of Visual Languages & Computing*, 20(3):131–144, 2009.

[84] P. Turaga, S. Biswas, and R. Chellappa. The role of geometry in age estimation. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 946–949. IEEE, 2010.

[85] A. J. O'Toole, T. Price, T. Vetter, J. C. Bartlett, and V. Blanz. 3d shape and 2d surface textures of human faces: The role of "averages" in attractiveness and age. *Image and Vision Computing*, 18(1):9–19, 1999.

[86] T. Wu, P. Turaga, and R. Chellappa. Age estimation and face verification across aging using landmarks. *IEEE Transactions on Information Forensics and Security*, 7(6):1780–1788, 2012.

[87] J.-C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep cnn features. *arXiv preprint arXiv:1508.01722*, 2015.

[88] X. Geng, C. Yin, and Z. Zhou. Facial age estimation by learning from label distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2401–2412, 2013.

[89] A. Saxena, S. Sharma, and V. K. Chaurasiya. Neural network based human age-group estimation in curvelet domain. *Procedia Computer Science*, 54:781–789, 2015.

[90] S. N. Kohail. Using artificial neural network for human age estimation based on facial images. In *International Conference on Innovations in Information Technology*, pages 215–219. IEEE, 2012.

[91] P. Thukral, K. Mitra, and R. Chellappa. A hierarchical approach for human age estimation. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1529–1532. IEEE, 2012.

[92] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[93] V. M. Patel, T. Wu, S. Biswas, P. J. Phillips, and R. Chellappa. Dictionary-based face recognition under variable lighting and pose. *IEEE Transactions on Information Forensics and Security*, 7(3):954–965, 2012.

[94] V. M. Patel, Y.-C. Chen, R. Chellappa, and P. J. Phillips. Dictionaries for image and video-based face recognition. *Journal of the Optical Society of America A*, 31(5):1090–1103, 2014.

[95] N. Shroff, P. Turaga, and R. Chellappa. Video precis: Highlighting diverse aspects of videos. *IEEE Transactions on Multimedia*, 12(8):853–868, 2010.

[96] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 545–552, 2011.

[97] X. D. Cao, Y. C. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2887–2894, 2012.

[98] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa. Design of non-linear kernel dictionaries for object recognition. *IEEE Transactions on Image Processing*, 22(12):5123–5135, 2012.

[99] P. J. Phillips, P. J. Flynn, J. R. Beveridge, W. T. Scruggs, A. J. OToole, D. Bolme, K. W. Bowyer, B. A. Draper, G. H. Givens, and Y. M. Lui. Overview of the multiple biometrics grand challenge. In *Advances in Biometrics*, pages 705–714. Springer, 2009.

[100] National institute of standards and technology: Face and ocular challenge series, http://www.nist.gov/itl/iad/ig/focs.cfm.

[101] Y.-C. Chen, V. M. Patel, S. Shekhar, R. Chellappa, and P. J. Phillips. Video-based face recognition via joint sparse representation. In *IEEE conference on Automatic Face and Gesture Recognition*, 2013.

[102] D. A. Ross, J. W. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008.

[103] P. Turaga, A. Veeraraghavan, and R. Chellappa. Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[104] R. P. Wang and X. L. Chen. Manifold discriminant analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 429–436, 2009.

[105] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on Faces in Real-Life Images: Detection, Alignment, and Recognition*, 2008.

[106] A. Asthana, S. Zafeiriou, S. Y. Cheng, and M. Pantic. Incremental face alignment in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1859–1866, 2014.

[107] H. T. Wang, S. Z. Li, and Y. S. Wang. Face recognition under varying lighting conditions using self quotient image. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 819–824. IEEE, 2004.

[108] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *In European Conference on Computer Vision*, pages 143–156. 2010.

[109] H. X. Li, G. Hua, Z. Lin, J. Brandt, and J. C. Yang. Probabilistic elastic matching for pose variant face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3499–3506, 2013.

[110] X. D. Cao, D. Wipf, F. Wen, G. Q. Duan, and J. Sun. A practical transfer learning algorithm for face verification. In *IEEE International Conference on Computer Vision*, pages 3208–3215. IEEE, 2013.

[111] National institute of standards and technology: Multiple biometric grand challenge, http://www.nist.gov/itl/iad/ig/mbgc.cfm.

[112] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2911–2918. IEEE, 2012.

[113] R. Beveridge, H. Zhang, B. Draper, P. Flynn, Z. Feng, P. Huber, J. Kittler, Z. Huang, S. Li, Y. Li, M. Kan, R. Wang, S. Shan, X. Chen, H. Li, G. Hua, V. Struc, J. Krizaj, C. Ding, D. Tao, and J. Phillips. Report on the fg 2015 video person

recognition evaluation. *IEEE International Conf. on Face and Gesture Recognition*, 2015.

[114] J.-C. Chen, S. Sankaranarayanan, V. M. Patel, and R. Chellappa. Unconstrained face verification using Fisher vectors computed from frontalized faces. In *IEEE International Conference on Biometrics: Theory, Applications and Systems*, 2015.

[115] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[116] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[117] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *arXiv preprint arXiv:1506.02640*, 2015.

[118] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. Ssd: Single shot multibox detector. *arXiv preprint arXiv:1512.02325*, 2015.

[119] National institute of standards and technology (NIST): IARPA Janus benchmark-a performance report.

[120] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1685–1692, June 2014.

[121] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.

[122] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[123] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv preprint arXiv:1502.01852*, 2015.

[124] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[125] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, pages 3320–3328, 2014.

[126] M. Long and J. Wang. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

[127] S. Sankaranarayanan, A. Alavi, and R. Chellappa. Triplet similarity embedding for face verification, 2016.

[128] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[129] J. Cheney, B. Klein, A. K. Jain, and B. F. Klare. Unconstrained face detection: State of the art baseline and challenges. In *International Conference on Biometrics*, 2015.

[130] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Number UM-CS-2010-009, 2010.

[131] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Convolutional channel features. In *IEEE International Conference on Computer Vision*, 2015.

[132] J. Yan, X. Zhang, Z. Lei, and S. Z. Li. Face detection by structural models. *Image and Vision Computing*, 32(10):790 – 799, 2014. Best of Automatic Face and Gesture Recognition 2013.

[133] S. Liao, A. Jain, and S. Li. A fast and accurate unconstrained face detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[134] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang. Probabilistic elastic part model for unsupervised face detector adaptation. In *IEEE International Conference on Computer Vision*, pages 793–800, Dec 2013.

[135] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2879–2886. IEEE, 2012.

[136] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2930–2940, 2013.

[137] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *European Conference on Computer Vision*, pages 679–692. Springer, 2012.

[138] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010.

[139] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3444–3451, 2013.

[140] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.

[141] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *IEEE International Conference on Computer Vision*, ICCV '13, pages 1513–1520, Washington, DC, USA, 2013. IEEE Computer Society.

[142] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 532–539, June 2013.

[143] G. Tzimiropoulos and M. Pantic. Gauss-newton deformable part models for face alignment in-the-wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1851–1858, June 2014.

[144] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment. In *European Conference on Computer Vision ECCV*, pages 1–16, 2014.

[145] S. Zhu, C. Li, C. L. Chen, and X. Tang. Face alignment by coarse-to-fine shape searching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[146] D. Wang, C. Otto, and A. K. Jain. Face search at scale: 80 million gallery. *arXiv preprint arXiv:1507.07242*, 2015.

[147] A. RoyChowdhury, T.-Y. Lin, S. Maji, and E. Learned-Miller. One-to-many face recognition with bilinear cnns. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[148] J. Yang, P. Ren, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation network for video face recognition. *arXiv preprint arXiv:1603.05474*, 2016.

[149] I. Masi, A. T. Tran, J. T. Leksut, T. Hassner, and G. Medioni. Do we really need to collect millions of faces for effective face recognition? *arXiv preprint arXiv:1603.07057*, 2016.

[150] W. AbdAlmageed, Y. Wu, S. Rawls, S. Harel, T. Hassne, I. Masi, J. Choi, J. Lekust, J. Kim, P. Natarajana, R. Nevatia, and G. Medioni. Face recognition using deep multi-pose representations. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[151] S. Sankaranarayanan, A. Alavi, C. Castillo, and R. Chellappa. Triplet Probabilistic Embedding for Face Verification and Clustering. *ArXiv e-prints*, April 2016.

[152] N. Crosswhite, J. Byrne, O. M. Parkhi, C. Stauffer, Q. Cao, and A. Zisserman. Template adaptation for face verification and identification. *arXiv preprint arXiv:1603.03958*, 2016.

[153] Y. Sun, D. Liang, X. Wang, and X. Tang. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015.

[154] C. Ding and D. Tao. Robust face recognition via multimodal deep face representation. *arXiv preprint arXiv:1509.00244*, 2015.

[155] L. Wolf, T. Hassner, and Y. Taigman. The one-shot similarity kernel. In *International Conference on Computer Vision*, pages 897–902. IEEE, 2009.

[156] Y. Fu, G. Guo, and T. Huang. Age synthesis and estimation via faces: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):1955–1976, 2010.

[157] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.

[158] A. Gallagher and T. Chen. Understanding images of groups of people. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[159] E. Eidinger, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security*, 9(12):2170–2179, 2014.

[160] H. Han, C. Otto, and A. K. Jain. Age estimation from face images: Human vs. machine performance. In *2013 International Conference on Biometrics (ICB)*, pages 1–8. IEEE, 2013.

[161] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *IEEE International Conference on Neural Networks*, pages 586–591, 1993.

[162] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[163] R. Rothe, R. Timofte, and L. V. Gool. Dex: Deep expectation of apparent age from a single image. In *ICCV, ChaLearn Looking at People workshop*, December 2015.

[164] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*, pages 675–678, 2014.

[165] G. Levi and T. Hassner. Age and gender classification using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) workshops*, June 2015.

[166] K. Luu, K. Ricanek, T. D. Bui, and C. Y. Suen. Age estimation using active appearance models and support vector machine regression. In *IEEE International Conference on Biometrics: Theory, Applications, and Systems.*, pages 1–5, Sept 2009.

[167] J. Ylioinas, A. Hadid, X. Hong, and M. Pietikäinen. Age estimation using local binary pattern kernel density estimate. In Alfredo Petrosino, editor, *Image Analysis and Processing (ICIAP)*, volume 8156 of *Lecture Notes in Computer Science*, pages 141–150. Springer Berlin Heidelberg, 2013.

[168] X. Geng, Z.-H. Zhou, and K. Smith-Miles. Automatic age estimation based on facial aging patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2234–2240, 2007.

[169] K. Chen, S. Gong, T. Xiang, and C.C. Loy. Cumulative attribute space for age and crowd density estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2467–2474, June 2013.

[170] M.Y. El Dib and M. El-Saban. Human age estimation using enhanced bio-inspired features (ebif). In *IEEE International Conference on Image Processing (ICIP)*, pages 1589–1592, Sept 2010.

[171] L. Hong, D. Wen, C. Fang, and X. Ding. A new biologically inspired active appearance model for face age estimation by using local ordinal ranking. In *International Conference on Internet Multimedia Computing and Service*, ICIMCS '13, pages 327–330, New York, NY, USA, 2013. ACM.

[172] W.-L. Chao, J.-Z. Liu, and J.-J. Ding. Facial age estimation based on label-sensitive learning and age-oriented regression. *Pattern Recognition*, 46(3):628 – 641, 2013.

[173] D. Chen, G. Hua, F. Wen, and J. Sun. Supervised transformer network for efficient face detection. In *European Conference on Computer Vision*, pages 122–138. Springer, 2016.

[174] J.-C. Chen, J. Zheng, V. M Patel, and R. Chellappa. Fisher vector encoded deep convolutional features for unconstrained face verification. 2016.

[175] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, pages 87–102. Springer, 2016.