ABSTRACT

Title of dissertation: AUGMENTED DEEP REPRESENTATIONS FOR UNCONSTRAINED STILL/VIDEO-BASED FACE RECOGNITION

Jingxiao Zheng
Doctor of Philosophy, 2019

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

Face recognition is one of the active areas of research in computer vision and biometrics. Many approaches have been proposed in the literature that demonstrate impressive performance, especially those based on deep learning. However, unconstrained face recognition with large pose, illumination, occlusion and other variations is still an unsolved problem. Unconstrained video-based face recognition is even more challenging due to the large volume of data to be processed, lack of labeled training data and significant intra/inter-video variations on scene, blur, video quality, etc. Although Deep Convolutional Neural Networks (DCNNs) have provided discriminant representations for faces and achieved performance surpassing humans in controlled scenarios, modifications are necessary for face recognition in unconstrained conditions. In this dissertation, we propose several methods that improve unconstrained face recognition performance by augmenting the representation provided by the deep networks using correlation or contextual information in the data.

For unconstrained still face recognition, we present an encoding approach to combine the Fisher vector (FV) encoding and DCNN representations, which is called FV-DCNN. The feature maps from the last convolutional layer in the deep network are encoded by FV into a robust representation, which utilizes the correlation between facial parts within each face. A VLAD-based encoding method called VLAD-DCNN is also proposed as an extension. Extensive evaluations on three challenging face recognition datasets show that the proposed FV-DCNN and VLAD-DCNN perform comparable to or better than many state-of-the-art face verification methods.

For the more challenging video-based face recognition task, we first propose an automatic system and model the video-to-video similarity as subspace-to-subspace similarity, where the subspaces characterize the correlation between deep representations of faces in videos. In the system, a quality-aware subspace-to-subspace similarity is introduced, where subspaces are learned using quality-aware principal component analysis. Subspaces along with quality-aware exemplars of templates are used to produce the similarity scores between video pairs by a quality-aware principal angle-based subspace-to-subspace similarity metric. The method is evaluated on four video datasets. The experimental results demonstrate the superior performance of the proposed method.

To utilize the temporal information in videos, a hybrid dictionary learning method is also proposed for video-based face recognition. The proposed unsupervised approach effectively models the temporal correlation between deep representations of video faces using dynamical dictionaries. A practical iterative optimization algorithm is introduced to learn the dynamical dictionary. Experiments on three

video-based face recognition datasets demonstrate that the proposed method can effectively learn robust and discriminative representation for videos and improve the face recognition performance.

Finally, to leverage contextual information in videos, we present the Uncertainty-Gated Graph (UGG) for unconstrained video-based face recognition. It utilizes contextual information between faces by conducting graph-based identity propagation between sample tracklets, where identity information are initialized by the deep representations of video faces. UGG explicitly models the uncertainty of the contextual connections between tracklets by adaptively updating the weights of the edge gates according to the identity distributions of the nodes during inference. UGG is a generic graphical model that can be applied at only inference time or with end-to-end training. We demonstrate the effectiveness of UGG with state-of-the-art results on the recently released challenging Cast Search in Movies and IARPA Janus Surveillance Video Benchmark datasets.

AUGMENTED DEEP REPRESENTATIONS FOR UNCONSTRAINED STILL/VIDEO-BASED FACE RECOGNITION

by

Jingxiao Zheng

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

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor Joseph F. JaJa
Professor Behtash Babadi
Professor Vishal M. Patel
Professor Ramani Duraiswami (Dean's Representative)

Dedication

To my parents and grandparents

For their love, sacrifice and selflessness.

Acknowledgments

I am deeply grateful to all the people who helped and supported me through my Ph.D. study and made this dissertation possible.

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Rama Chellappa, for his continuous support during my Ph.D. study. Being a caring advisor, a dedicated researcher and a charismatic leader, he guided me to tackle challenging problems in research, inspired me by his creative ideas and encouraged me to overcome difficult times in life. I feel incredibly fortunate to have such an amazing advisor, without whom this dissertation will not be possible.

I am honored to have Professor Joseph JaJa, Professor Behtash Babadi, Professor Ramani Duraiswami and Professor Vishal Patel in my dissertation committee. I would like to thank them for serving in my committee and providing valuable feedback to improve the quality of this dissertation.

My sincere thanks goes to my internship mentor, Dr. S. Kevin Zhou, at Siemens Healthineers. His precious support and insightful guidance broadened my research horizons.

I would also like to thank all the members of Professor Chellappa's group and all my collaborators, especially Dr. Jun-Cheng Chen, Dr. Boyu Lu, Dr. Ruichi Yu and Dr. Carlos Castillo. A special mention goes to Dr. Jun-Cheng Chen who kindly guided me through my research career. I learned a lot from his rigorous scholarship and his passion for exploring.

Finally, I would like to express my deepest gratitude to my parents, grand-

parents, uncles and aunts who support me all the time in my life. In particular, I want to thank my girlfriend, Siqin Li, who always supports and encourages me throughout my preparation for this dissertation.

My research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2019-022600002. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Motivation

Face recognition is one of the most actively studied problems in computer vision and biometrics. It has a wide range of applications including visual surveillance, access control, etc. Basically, face recognition can be categorized into two tasks: face identification which matches a given face query to one of the identities in a pre-enrolled face gallery, and face verification focusing on deciding whether a pair of face queries belongs to the same identity. For both tasks, it is crucial to learn robust and discriminative representations for faces.

Recently, with the availability of powerful GPUs and large amounts of labeled training data, deep convolutional neural networks (DCNNs) have demonstrated impressive performances on many computer vision tasks such as object recognition [40, 57, 100], object detection [39, 83], face detection [68, 82] and semantic segmentation [18]. It has been shown that a DCNN model 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. As a result, DCNNs have also produced state-of-the-art results for face recognition as reported in [14,72,79,89,102].

However, the unconstrained face recognition problem with large pose, illumina-

tion, occlusion and other variations is still unsolved. Compared to still image-based face recognition, video-based face recognition is more challenging due to a much larger amount of data to be processed and significant intra/inter-class variations caused by motion blur, low video quality, occlusion, frequent scene changes, and unconstrained acquisition conditions.

Popular off-the-shelf DCNNs have provided discriminant representation for faces and overcome illumination and small pose variation already. But their performance on faces captured in unconstrained conditions or in videos is far from satisfying. To fill the performance gap between face recognition in controlled still-faces and unconstrained/video faces, training a specific model needs large amount of annotated data in similar domains which is very difficult and costly to collect.

In this dissertation, we focus on the theme of "augmented deep representations". We propose four examples of augmented deep representations in the following chapters. In order to produce more efficient and discriminative representations for face recognition, these methods augment deep representations extracted from the well-studied deep learning-based still face recognition approaches by utilizing additional information in the data.

1.2 Overview

In Chapter 2, we propose two augmentation methods of DCNN features, FV-DCNN and VLAD-DCNN, to handle large pose variations in unconstrained face recognition by encoding the spatially distributed deep representations from the last

convolutional layer and leveraging the spatial information in face images. For the challenging video-base face recognition problem, in Chapter 3 we build subspaces to leverage the correlation between deep representations of faces in the same set. It acts as an important component of the proposed automatic video-based face recognition system. Continuing on this topic, we exploit the temporal information in video faces by learning linear dynamical dictionaries from deep representations in Chapter 4. To utilize contextual information in videos, in Chapter 5, we further propose a graphical-model-based framework called Uncertainty-Gated Graph (UGG) for video-based face recognition to model the contextual connections between face tracklets. Identity information from deep representations is propagated in the adaptive graph built for each video. Finally, in Chapter 6, we conclude the dissertation and discuss possible future directions.

In the following sections of this chapter, we will introduce the proposed methods with more details.

1.3 Spatial Encoding of Deep Convolutional Features for Unconstrained Face Recognition

As discussed previously, the unconstrained still face recognition problem with large pose, illumination, occlusion and other variations is an unsolved problem. In still faces, discriminant facial landmarks like eyes, nose, mouth, ears are distributed at different spatial locations. Local features extracted around these landmarks only capture partial information from the face and need to be encoded into a more robust

representation. When the alignment of faces is poor due to extreme poses, there will be large spatial variations of these landmarks. DCNNs usually do not account for these spatial variations explicitly. As the last convolutional layer in many popular DCNN models is often followed by a simple average pooling layer so the spatial information is lost.

In computer vision, many methods have been proposed to extract local spatial features from images and encode them into high-dimensional features to handle large data variations and noise. Several approaches have combined feature encoding with deep learning and successfully improved performance. Gong *et al.* [37] extracted the multi-scale deep features followed by VLAD encoding [48] for feature encoding and demonstrated promising results for image retrieval and classification tasks. Cimpoi *et al.* [22] proposed a FV-DCNN approach to combine Fisher Vector (FV) [75] with DCNN features for texture recognition.

Motivated by the success of feature encoding and deep learning for various computer vision problems, in Chapter 2, we propose two augmentation methods which essentially leverage the spatial information on faces by combining feature encoding with DCNN representations for face recognition. We adopt a network architecture similar to the one proposed in [121] and encode FV-DCNN/VLAD-DCNN representations using the feature maps from the last convolutional layer of the network. The spatial information in the feature maps ignored by average pooling is incorporated by adding two additional spatial coordinate dimensions into the features for FV/VLAD encoding.

We evaluate FV-DCNN on two face verification datasets: Celebrities in Frontal-

Profile (CFP) dataset [90] and Labeled Face in the Wild (LFW) dataset [45]. We also evaluate the performance of VLAD-DCNN on two unconstrained face recognition datasets: IARPA Janus Benchmark A (IJB-A) [55] and its extension JANUS Challenge Set 2 (JANUS CS2). Extensive evaluations show that the proposed FV-DCNN and VLAD-DCNN perform comparable to or better than many state-of-the-art face recognition methods. Experiments also show that VLAD-DCNN works better than FV-DCNN because of the noisy second order statistics of DCNN features.

1.4 Quality-Aware Subspace Learning and Matching for Video-based Face Recognition

Next, we address the more challenging video-based face recognition problem. In video-based face recognition, the first challenge is that face representations must be robust to large within-subject variations in videos. The second challenge is how to efficiently aggregate a varying-length set of features into a fixed-size and unified representation, since each video contains different number of faces for each subject.

Since in videos, faces from the same subject are usually associated into a set by face association, there is correlation between faces in the same set and this information can be leveraged to improve the face recognition performance. Representative and discriminative models based on manifolds and subspaces have received attention for image set-based face recognition [109] [107]. These methods model sets of face features as manifolds or subspaces and use appropriate similarity metric for set-based identification and verification, without any external training data.

Following this direction, we propose an automatic system for unconstrained video-based face recognition in Chapter 3. The system first detects faces and facial landmarks. Then deep representations from detected faces are extracted using state-of-the-art DCNNs. Target faces from single-shot/multiple-shot videos are associated by tracking and association methods. Finally, we learn a subspace representation from each video template and match pairs of templates using principal angles-based subspace-to-subspace similarity metric on the subspace representations. Advantages of subspace-based methods include: 1) the subspace representation encodes the correlation between samples. Exploiting correlation between samples by subspaces help learn a more robust representation to capture variations in videos. 2) a fixed-size representation can be learned from an arbitrary number of video frames.

We evaluate our face recognition system on the challenging IARPA Benchmark B (IJB-B) [112] and IARPA Janus Surveillance Video Benchmark (IJB-S) [52] datasets, as well as the Multiple Biometric Grand Challenge (MBGC) [67] dataset and the Face and Ocular Challenge Series (FOCS) [69] dataset, and the results demonstrate that the proposed system achieves improved performance over other deep learning-based baselines and state-of-the-art approaches.

1.5 Hybrid Dictionary Learning and Matching for Video-based Face Verification

In the video-based face recognition method discussed above, we model the faces from each subject in a video as a set without temporal order. Sometimes faces

in videos are tracked into sequences by face trackers. So there is temporal correlation between faces from adjacent frames and we can exploit this temporal information for more robust face representations. For feature aggregation in sequential data, temporal deep learning models such as Recurrent Neural Network (RNN) can be applied. However, training these models needs large-scale labeled training data which is very expensive to collect in the context of video-based recognition. Linear Dynamical Systems (LDSs) play an important role in representing sequential data. A wide variety of spatio-temporal signals has been modeled as realizations of LDSs [101]. On the other hand, dictionary learning methods model the data generatively without pretraining on external data, which is an advantage compared with other RNN-based approaches. Traditional dictionary learning methods are specifically designed for still images. But the idea can be easily incorporated into an LDS model as well.

Therefore, by combining deep learning, sparse representations and LDS models, in Chapter 4 we propose a hybrid dictionary learning and matching approach for unconstrained video-based face verification, in order to utilize the temporal information in the videos. The proposed method learns both structural and dynamical dictionaries from videos, where dynamical dictionaries and LDSs are jointly learned using the proposed Linear Dynamical Dictionary Learning (LDDL) algorithm. With the learned dictionaries, the similarity between videos is measured by subspace-to-subspace similarity, where the subspaces are spanned by the dictionaries and encode the correlation of the deep features in videos.

Experiments on three video-based face recognition datasets: Multiple Bio-

metric Grand Challenge (MBGC), Face and Ocular Challenge Series (FOCS) and IARPA Janus Benchmark A (IJB-A) [55] demonstrate that the proposed method can effectively learn robust and discriminative representation for videos and improve the face recognition performance.

1.6 Modeling Contextual Information by Graphical-Models for Video-based Face Recognition

For video-based face recognition, improving the recognition performance on faces with extreme variations is always challenging. An effective idea is to utilize some video contextual information, such as body appearance and spatial-temporal correlation between person instances, to propagate the identity information from high-quality faces to low-quality ones. It has been explored using graph-based approaches [30, 46, 92] in which graphs are constructed with nodes to represent one or more frames (tracklets) of faces and edges to connect tracklets. However, misleading connections in the graph may propagate erroneous information.

To address the problem, in Chapter 5 we propose a conditional random field-based framework called Uncertainty-Gated Graph (UGG) to built more reliable connections using contextual information. In UGG, the identity information of tracklets from their deep representations is propagated through the connections, that are adaptively updated by the connected tracklets. We model two types of contextual connections separately, which allows our model to consider different contextual information in challenging conditions, and leads to improved performance.

The proposed method is evaluated on two challenging datasets, the Cast Search in Movies (CSM) dataset [46] and the IARPA Janus Surveillance Video Benchmark (IJB-S) dataset with superior performance compared to existing methods.

1.7 Contributions

- In Chapter 2, we propose two approaches that combine spatial feature encoding and DCNN representation for unconstrained face recognition.
  - We propose FV-DCNN which encodes the spatially augmented feature map from the last convolutional layer of a DCNN model by Fisher vector.
  - We propose VLAD-DCNN which encodes the spatially augmented feature map by VLAD encoding, as an extension of FV-DCNN.
  - We evaluate FV-DCNN with VLAD-DCNN on several benchmark face datasets and show that for unconstrained face recognition, VLAD-DCNN works better than FV-DCNN because of the noisy second order statistics of DCNN features.
- In Chapter 3, we propose an automatic video-based face recognition system with components including face/fiducial detection, face association, and face recognition.
  - To exploit the correlation in face sets, we propose a quality-aware subspace learning approach for face feature aggregation.
  - We compute the video set-to-set similarity using a subspace-based simi-

larity metric for video-based face recognition. A variance-aware subspace-to-subspace similarity metric is also proposed.

- In Chapter 4, we propose a hybrid dictionary learning and matching approach for video-based face verification.
  - We model the temporal correlation between DCNN features in videos using dynamical dictionaries.
  - A practical iterative optimization algorithm, Linear Dynamical Dictionary Learning (LDDL), is proposed to learn the dynamical dictionary.
  - We compute the video-to-video similarity by subspace-to-subspace similarity where the subspaces are spanned by the learned structural and dynamical dictionaries.
- In Chapter 5, we propose the UGG model to leverage contextual information in videos for video-based face recognition.
  - We explicitly model the uncertainty of connections between tracklets using uncertainty gates over graph edges.
  - The tracklets and gates in the graph are updated jointly and possible connection errors can be corrected during inference.
  - We utilize both positive and negative connections for information propagation.
  - The proposed method is efficient and flexible. It can either be used at inference time without supervision, or be considered as a trainable module

for supervised and semi-supervised training.

– We achieve state-of-the-art results on two challenging datasets, the CSM dataset and the IJB-S dataset.

Chapter 2: Fisher Vector/VLAD Encoded Deep Convolutional Features for Unconstrained Face Recognition

2.1 Introduction

Learning invariant and discriminative features from images and videos is one of the central goals of research in many computer vision tasks such as object recognition and face recognition. Many approaches have been proposed in the literature that extract over-complete and high-dimensional features from images to handle large data variations and noise. For instance, the high-dimensional multi-scale Local Binary Pattern (LBP) [12] representation extracted from local patches around facial landmarks is reasonably effective for face recognition. Face representations based on Fisher vector (FV) have also shown to be effective for face recognition problems [16, 71, 94]. Other feature encoding methods that have been successfully used in computer vision applications include Bag-of-Visual-Words (BoVW) model [24], Vector of Locally Aggregated Descriptor (VLAD) [48] and Super Vector Coding [131].

In the era of deep learning, many approaches have combined deep learning with other feature encoding methods to further improve performance. Gong *et al.* [37]

extracted multi-scale deep features followed by VLAD for feature encoding and demonstrated promising results for image retrieval and classification tasks. Cimpoi *et al.* [22] proposed a FV-DCNN approach to combine FV with DCNN features for texture recognition.

Motivated by the success of combining feature encoding and deep learning in various computer vision problems, we propose two face recognition methods which essentially apply feature encoding method on DCNN representations to leverage the spatial information for face recognition. An overview of the proposed FV-DCNN and VLAD-DCNN methods for face recognition is shown in Figures 2.1 and 2.2 respectively. We adopt a network architecture similar to the one proposed in [121] that has demonstrated good performance for face recognition. The DCNN model builds a 15-layer deep architecture for convolutional neural network by stacking small filters (*i.e.* 3×3) together as VGGNet [95] and is trained using the CASIA-WebFace dataset [121] of 10,548 subjects. FV-DCNN/VLAD-DCNN features are encoded by the feature maps coming out of the last convolutional layer of the network. These feature maps contain spatial information ignored by average pooling. Unlike some of the previous approaches [22] and [37], the spatial information is also encoded by adding two spatial coordinate dimensions into the features. In our method, we also use the average pooling features from the last convolutional layer and output features from the fully-connected layer. Discriminative metrics learned from the training set are applied on these features to compute similarities between feature pairs.

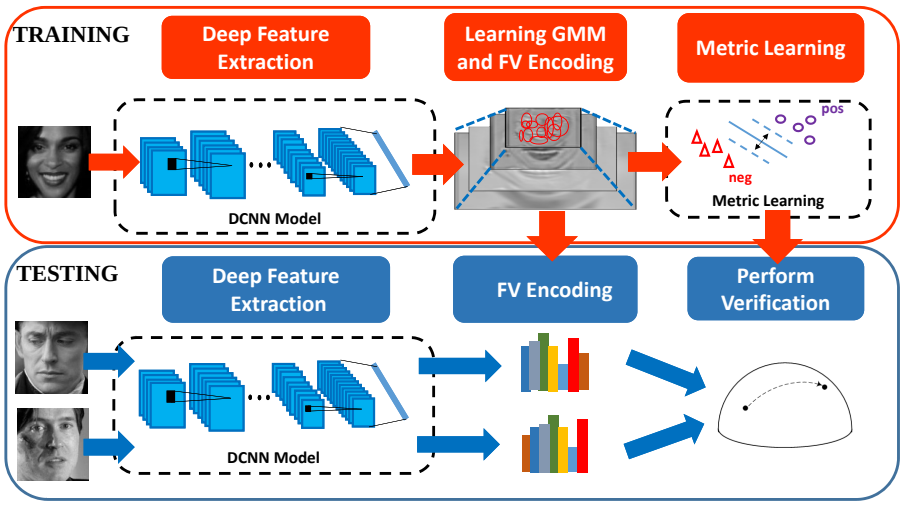


Figure 2.1: An overview of the proposed FV-DCNN representation for unconstrained face recognition.

2.2 Method

The system pipelines for FV-DCNN and VLAD-DCNN are very similar. In the training phase, each training image is first passed through a pre-trained DCNN model to extract the convolutional features *conv* from the last convolutional layer, the average pooled features *pool* from the last convolutional layer and the output features *fc* of the fully-connected layer. We learn the Gaussian mixture model over *conv* for FV-DCNN. The K-means clustering algorithm is applied on them for VLAD-DCNN. We then perform the corresponding feature encoding over these local deep features to generate the encoded representations *conv_fv* and *conv_vlad*. Finally, we learn the metric from these features.

In the testing phase, we extract the DCNN features *conv*, *pool* and *fc* and use the learned GMM/K-means to perform the corresponding feature encoding. We

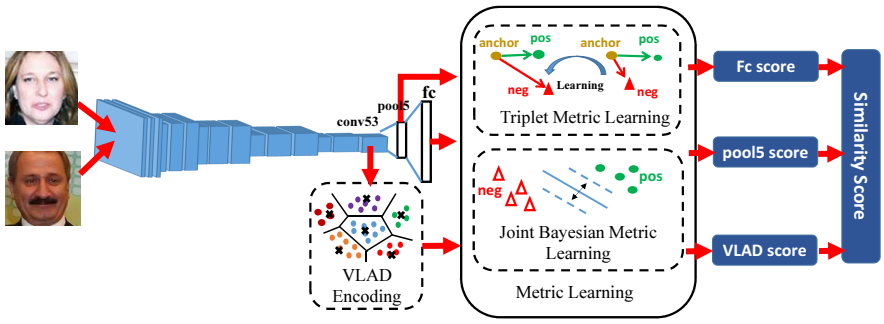


Figure 2.2: An overview of the proposed VLAD-DCNN framework to combine the global average pooling, fully-connected layer features and VLAD features for unconstrained face recognition.

then apply the learned metric to compute the similarity scores. We describe the details of each of these components in the following sections.

2.2.1 Deep Face Representation

The DCNN model proposed in [14] is used for FV-DCNN. VLAD-DCNN further exploits an improved version with 15 convolutional layers, 5 pooling layers and 2 fully connected layers as shown in Table 2.1. Both models are trained using the CASIA-WebFace dataset [121] with cross-entropy loss. We use the parametric ReLU (PReLU) [41] as the nonlinear activation function. The network input is $100 \times 100 \times 1$ gray-scale image for FV-DCNN and $100 \times 100 \times 3$ RGB image for VLAD-DCNN.

We use the output of conv52/conv53 layer as the *conv* features for FV-DCNN and VLAD-DCNN respectively. The *pool* features are the output of pool5 layer and the *fc* features are the output of fc6 layer.

Name	Type	Filter Size/Ouput/Stride	#Params
Conv11	convolution	3×3 / 32 / 1	0.28K
Conv12	convolution	3×3 / 64 / 1	18K
Conv13	convolution	3×3 / 64 / 1	36K
Pool1	max pooling	2×2 / 2	
Conv21	convolution	3×3 / 64 / 1	36K
Conv22	convolution	3×3 / 128 / 1	72K
Conv23	convolution	3×3 / 128 / 1	144K
Pool2	max pooling	2×2 / 2	
Conv31	convolution	3×3 / 96 / 1	108K
Conv32	convolution	3×3 / 192 / 1	162K
Conv33	convolution	3×3 / 192 / 1	324K
Pool3	max pooling	2×2 / 2	
Conv41	convolution	3×3 / 128 / 1	216K
Conv42	convolution	3×3 / 256 / 1	288K
Conv43	convolution	3×3 / 256 / 1	576K
Pool4	max pooling	2×2 / 2	
Conv51	convolution	3×3 / 160 / 1	360K
Conv52	convolution	3×3 / 320 / 1	450K
Conv53	convolution	3×3 / 320 / 1	900K
Pool5	avg pooling	7×7 / 1	
Dropout	dropout (40%)		
Fc6	fully connection	10548	3305K
Cost	softmax		
total			6995K

Table 2.1: The architecture of DCNN model for VLAD-DCNN.

2.2.2 Feature Encoding

Since the *pool* features are the average of the *conv* features from the last convolutional layer, they capture global discriminative information with less noise due to the average pooling operation. Each entry of the *fc* features shows how the input image looks like the corresponding person in the external training set. Different positions in *conv* feature maps correspond to different parts of the face. Even though the receptive fields of high level convolutional layers are largely overlapped, especially for deep networks, spatial information is still preserved in *conv* feature maps. Therefore, we use two feature encoding methods to incorporate this important information from a feature map into a more discriminative feature.

2.2.2.1 Fisher Vector Encoding

The Fisher Vector is a bag-of-visual-word approach which encodes a large set of local features into a high-dimensional vector according to the parametric generative model fitted for the features. The FV representation is computed by encoding the local features with the derivatives of the log-likelihood of the learned model with respect to model parameters. Similar to [75], we use a GMM in our work. The first-and second-order statistics of the features with respect to each component for

the FV representation are computed as follows:

$$\mathbf{\Phi}_{ik}^{(1)} = \frac{1}{N\sqrt{w_k}} \sum_{p=1}^{N} \alpha_k(\mathbf{v}_p) \left(\frac{\mathbf{v}_{ip} - \boldsymbol{\mu}_{ik}}{\boldsymbol{\sigma}_{ik}} \right), \tag{2.1}$$

$$\mathbf{\Phi}_{ik}^{(2)} = \frac{1}{N\sqrt{2w_k}} \sum_{p=1}^{N} \alpha_k(\mathbf{v}_p) \left(\frac{(\mathbf{v}_{ip} - \boldsymbol{\mu}_{ik})^2}{\boldsymbol{\sigma}_{ik}^2} - 1 \right), \tag{2.2}$$

$$\alpha_k(\mathbf{v}_p) = \frac{w_k \exp[-\frac{1}{2}(\mathbf{v}_p - \boldsymbol{\mu}_k)^T \mathbf{\Sigma}_k^{-1}(\mathbf{v}_p - \boldsymbol{\mu}_k)]}{\sum_i^K w_i \exp[-\frac{1}{2}(\mathbf{v}_p - \boldsymbol{\mu}_i)^T \mathbf{\Sigma}_i^{-1}(\mathbf{v}_p - \boldsymbol{\mu}_i)]}, \tag{2.3}$$

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. These parameters are learned from the training data using the EM algorithm. $\alpha_k(\mathbf{v}_p)$ is the posterior of $\mathbf{v}_p$ belonging to the kth mixture component. The FV representation $\mathbf{\Phi}(\mathbf{I})$ of an image $\mathbf{I}$ is obtained by concatenating all the $\mathbf{\Phi}_k^{(1)}$s and $\mathbf{\Phi}_k^{(2)}$s into a high-dimensional vector as

$$\mathbf{\Phi}(\mathbf{I}) = \left[(\mathbf{\Phi}_1^{(1)})^T, (\mathbf{\Phi}_1^{(2)})^T, \ldots, (\mathbf{\Phi}_K^{(1)})^T, (\mathbf{\Phi}_K^{(2)})^T \right]^T \tag{2.4}$$

whose dimensionality is $D = 2Kd$ where K is the number of mixture components, and d is the dimensionality of the local feature vector where we use $d = 322$ in this work.

2.2.2.2 VLAD Encoding

VLAD is a feature encoding method introduced in [49]. It encodes a set of local features into a high-dimensional vector using the clustering centers provided by methods like the K-means algorithm. For the kth cluster center $\boldsymbol{\mu}_k$, the corre-

sponding VLAD feature is calculated as the sum of the residuals as

$$\mathbf{v}_k = \sum_{i=1}^{N} \alpha_{ik}(\mathbf{x}_i - \boldsymbol{\mu}_k) \tag{2.5}$$

where $\{\mathbf{x}_i\}$ is the set of local features from an image I, α_{ik} is the association of data $\mathbf{x}_i$ to $\boldsymbol{\mu}_k$ with $\alpha_{ik} \geq 0$ and $\sum_{k=1}^{K} \alpha_{ik} = 1$. For hard association, we simply find the nearest neighbor of $\mathbf{x}_i$ among centers $\{\boldsymbol{\mu}_k\}$. As a result,

$$\alpha_{ik} = \begin{cases} 1 & \text{if } \|\mathbf{x}_i - \boldsymbol{\mu}_k\|_2 \leq \|\mathbf{x}_i - \boldsymbol{\mu}_l\|_2 \; \forall \; l \neq k \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

Then, the overall VLAD feature $\boldsymbol{\Phi}(I)$ for image I is stacked by the residuals for each center as

$$\boldsymbol{\Phi}(I) = \left[\mathbf{v}_1^T, \ldots, \mathbf{v}_K^T\right]^T. \tag{2.7}$$

VLAD encoding only involves the K-means clustering procedure and a nearest neighbor procedure (for hard assignment) to a cluster which can be done efficiently using a k-d tree. After finding the nearest neighbor for each feature, the encoding step is simply a summation of the feature residues.

2.2.2.3 Spatial Augmentation and Spatial Encoding

As shown in [94], spatially encoded local features are useful for face recognition. Thus, for both FV-DCNN and VLAD-DCNN, we augment the original *conv* features with the normalized x and y coordinates as $\left[\mathbf{f}_{xy}^T, x/w - 0.5, y/h - 0.5\right]^T$, where $\mathbf{f}_{xy}$ is the DCNN descriptor at (x, y), and w and h are the width and height of the *conv* feature map, respectively. By adding the two augmented dimensions, the clustering

method will not only cluster the training features in the feature space, but also consider their spatial relationships. The features that are closer in spatial domain will be more likely to be clustered together. Features that are far away will be more likely to be assigned to different clusters.

To balance the strength of appearance and spatial features, we take the square root and perform L_2 normalization on appearance features before augmenting spatial features. Moreover, we introduce an encoding scheme for FV-DCNN called "spatial encoding". Instead of using the original posterior (2.3), by spatial encoding, we enforce the feature to be encoded by its neighborhood, which is defined as

$$\tilde{\alpha}_k(\mathbf{v}_p) = \frac{w_k \exp[\frac{1}{2}(\tilde{\mathbf{v}}_p - \tilde{\boldsymbol{\mu}}_k)^T \tilde{\boldsymbol{\Sigma}}^{-1}(\tilde{\mathbf{v}}_p - \tilde{\boldsymbol{\mu}}_k)]}{\sum_i^K w_i \exp[\frac{1}{2}(\tilde{\mathbf{v}}_p - \tilde{\boldsymbol{\mu}}_i)^T \tilde{\boldsymbol{\Sigma}}^{-1}(\tilde{\mathbf{v}}_p - \tilde{\boldsymbol{\mu}}_i)]}, \tag{2.8}$$

where $\tilde{\boldsymbol{\mu}}_k$ and $\tilde{\boldsymbol{\Sigma}}_k$ are the mean and covariance of the two-dimensional spatial features for the kth Gaussian. The new posterior only considers the spatial distance between Gaussians and dense features, instead of the distance calculated among all dimensions. Spatial encoding improves the performance with well aligned images and reliable spatial information.

2.2.3 Metric Learning

After obtaining the encoded features, it is important to learn a similarity metric that is as discriminative as possible. There are many metric learning approaches in the literature [9,11,63,86,87]. In this work, we mainly focus on learning two kinds of metrics based on triplet distance embedding method and the Joint Bayesian (JB) method.

2.2.3.1 Triplit Distance Embedding

The triplet distance embedding has been widely used in the literature for different applications [87]. This embedding is obtained by solving the following optimization problem

$$\underset{\mathbf{W}}{\operatorname{argmin}} \sum_{\mathbf{x}_a,\mathbf{x}_p,\mathbf{x}_n \in \mathbb{T}} \max\{0, \alpha + (\mathbf{x}_a - \mathbf{x}_p)^T \mathbf{W}^T \mathbf{W}(\mathbf{x}_a - \mathbf{x}_p) - (\mathbf{x}_a - \mathbf{x}_n)^T \mathbf{W}^T \mathbf{W}(\mathbf{x}_a - \mathbf{x}_n)\}, \tag{2.9}$$

where $\mathbf{x}_a$, $\mathbf{x}_p$ and $\mathbf{x}_n$ are the anchor feature, positive feature and negative feature in the training triplet set $\mathbb{T}$, respectively. The goal of this embedding is to maximize the gap of the Euclidean distance between the positive and negative pairs with the same anchor in a triplet in the embedded space. The optimization problem can be solved using the Stochastic Gradient Descent (SGD) method and the corresponding update step is given by

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta \mathbf{W}_t [(\mathbf{x}_a - \mathbf{x}_p)(\mathbf{x}_a - \mathbf{x}_p)^T + (\mathbf{x}_a - \mathbf{x}_n)(\mathbf{x}_a - \mathbf{x}_n)^T] \tag{2.10}$$

when the update criterion $\alpha + (\mathbf{x}_a - \mathbf{x}_p)^T \mathbf{W}_t^T \mathbf{W}_t (\mathbf{x}_a - \mathbf{x}_p) - (\mathbf{x}_a - \mathbf{x}_n)^T \mathbf{W}_t^T \mathbf{W}_t (\mathbf{x}_a - \mathbf{x}_n) > 0$ is met. Here, we use a hard negative mining strategy introduced in [87]. Given any anchor feature, the negative feature is chosen as the closest feature in the embedded space to the anchor feature among a random subset of the negative candidates, which is

$$\mathbf{x}_n = \underset{\mathbf{x} \in \mathcal{C}(\mathbf{x}_a)}{\operatorname{argmin}} \|\mathbf{x}_a - \mathbf{x}_n\|_2, \tag{2.11}$$

where $\mathcal{C}(\mathbf{x}_a)$ is a random subset of the negative candidates of anchor $\mathbf{x}_a$. Given a testing pair $\mathbf{x}_i$ and $\mathbf{x}_j$, the similarity score is the squared Euclidean distance between

two features in the embedded space, which is

$$s(i,j) = \|\mathbf{W}(\mathbf{x}_i - \mathbf{x}_j)\|_2^2 = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{W}^T \mathbf{W}(\mathbf{x}_i - \mathbf{x}_j). \tag{2.12}$$

2.2.3.2 Joint Bayesian Approach

Another metric learning method we use is the JB approach, which has been widely used in the literature of face verification [9, 11]. We directly optimize the JB distance measure in a large-margin framework and update the model parameters using SGD as follows

$$\arg\min_{\mathbf{W},\mathbf{V},b} \sum_{i,j} \max\{1 - y_{ij}(b - (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{W}^T \mathbf{W}(\mathbf{x}_i - \mathbf{x}_j) + 2\mathbf{x}_i^T \mathbf{V}^T \mathbf{V}\mathbf{x}_j), 0\} \tag{2.13}$$

where $\mathbf{W}$ and $\mathbf{V} \in \mathbb{R}^{d \times D}$ with d and D as the dimensionality before and after dimension reduction. $b \in \mathbb{R}$ is the 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. Then, one can update $\mathbf{W}$ and $\mathbf{V}$ using the SGD method. The update equations are given as follows:

$$\begin{aligned}
\mathbf{W}_{t+1} &= \begin{cases} \mathbf{W}_t, & \text{if } y_{ij}(b_t - d_{\mathbf{W}_t,\mathbf{V}_t}(\mathbf{x}_i, \mathbf{x}_j)) > 1 \\ \mathbf{W}_t - \gamma y_{ij} \mathbf{W}_t \mathbf{\Gamma}_{ij}, & \text{otherwise,} \end{cases} \\
\mathbf{V}_{t+1} &= \begin{cases} \mathbf{V}_t, & \text{if } y_{ij}(b_t - d_{\mathbf{W}_t,\mathbf{V}_t}(\mathbf{x}_i, \mathbf{x}_j)) > 1 \\ \mathbf{V}_t + 2\gamma y_{ij} \mathbf{V}_t \mathbf{\Lambda}_{ij}, & \text{otherwise,} \end{cases} \\
b_{t+1} &= \begin{cases} b_t, & \text{if } y_{ij}(b_t - d_{\mathbf{W}_t,\mathbf{V}_t}(\mathbf{x}_i, \mathbf{x}_j)) > 1 \\ b_t + \gamma_b y_{ij}, & \text{otherwise,} \end{cases}
\end{aligned} \tag{2.14}$$

where $d_{\mathbf{W},\mathbf{V}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{W}^T \mathbf{W}(\mathbf{x}_i - \mathbf{x}_j) - 2\mathbf{x}_i^T \mathbf{V}^T \mathbf{V}\mathbf{x}_j$, $\mathbf{\Gamma}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T$, $\mathbf{\Lambda}_{ij} = \mathbf{x}_i \mathbf{x}_j^T + \mathbf{x}_j \mathbf{x}_i^T$ and γ is the learning rate for $\mathbf{W}$ and $\mathbf{V}$, and γ_b for the bias

b. We use the identity matrix to initialize both $\mathbf{W}$ and $\mathbf{V}$ if $d = D$. Otherwise, the projection matrix $\mathbf{P} \in \mathbb{R}^{d \times D}$ for dimension reduction is used for initialization. Both $\mathbf{W}$ and $\mathbf{V}$ are updated only when the constraints are violated. Given a testing pair $\mathbf{x}_i$ and $\mathbf{x}_j$, the similarity score is calculated as

$$s(i,j) = b - (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{W}^T \mathbf{W} (\mathbf{x}_i - \mathbf{x}_j) + 2\mathbf{x}_i^T \mathbf{V}^T \mathbf{V} \mathbf{x}_j. \tag{2.15}$$

2.2.4 Fusion

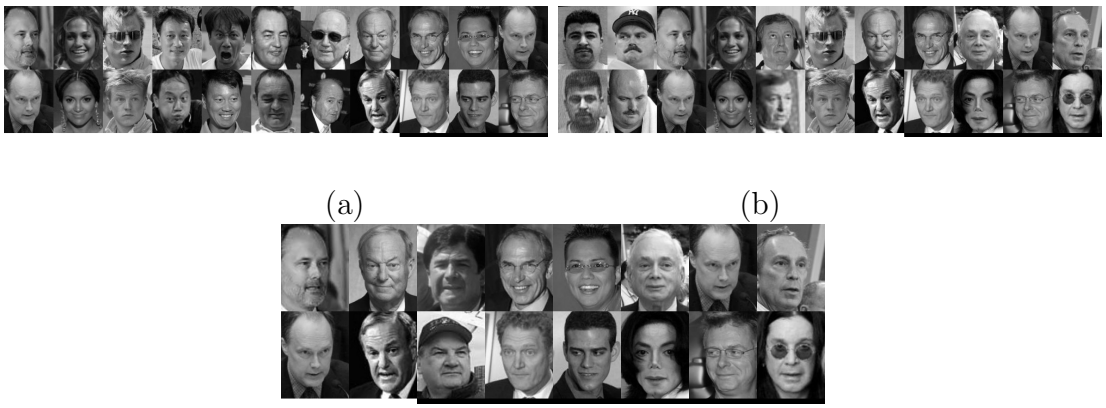


Figure 2.3: Errors made by different features in Split 10 of the LFW dataset. (a) *conv_fv* errors . (b) *pool* errors. (c) *conv_fv* + *pool* errors. Errors are significantly reduced when *conv_fv* and *pool* features are fused for verification.

In the experiments, we observe that the error patterns are different between pool5 and FV-DCNN features. Figure 2.3 (a) shows the errors made in Split 10 of the LFW dataset by *conv_fv* and Figure 2.3 (b) shows the errors made by the *pool* features. It is interesting to see how much the error is reduced when the *conv_fv* and *pool* features scores are fused. This can be seen by comparing the errors shown

in Figure 2.3 (c) with Figure 2.3 (a) and (b) where Figure 2.3 (c) is obtained by a linear combination of similarity scores from *conv_fv* and *pool*. Similarly, we apply score level fusion on *pool*, *fc* and *conv_vlad* features for VLAD-DCNN by a linear combination.

2.3 Experiments

We evaluate the proposed FV-DCNN on two face verification datasets: the Celebrities in Frontal-Profile (CFP) Dataset [90] and the Labeled Faces in the Wild (LFW) dataset [45]. The algorithm is evaluated using various metrics, including the ROC curves, equal error rate (EER), area under curve (AUC), and accuracy based on the test protocols defined for each dataset.

We also evaluate the performance of VLAD-DCNN on the IARPA Janus Benchmark A (IJB-A) [55] and its extension, JANUS Challenge Set 2 (JANUS CS2) unconstrained face recognition datasets.

2.3.1 Datasets

CFP: The CFP dataset focuses on the unconstrained frontal to profile face verification protocol where most profile faces are in extreme poses. Sample face pairs are shown in Figure 2.4. The dataset contains 500 subjects, and each subject contains 10 frontal and 4 profile images. The CFP dataset consists of 20 splits in total, 10 for frontal-to-frontal and the other 10 for frontal-to-profile face verification tasks. Each split has 350 same and 350 different pairs, respectively.

Figure 2.4: Sample image pairs from the CFP dataset where our method is able to successfully verify the pairs whereas both FV and DCNN-based methods fail.

LFW: The standard protocol for the face verification task of the LFW dataset defines 3,000 positive pairs and 3,000 negative pairs in total. The pairs are further split into 10 disjoint subsets for cross validation, and each subset consists of 300 same and 300 different pairs. It contains 7,701 images of 4,281 subjects.

IJB-A and JANUS CS2: Both IJB-A and JANUS CS2 datasets contain 500 subjects with 5,397 images and 2,042 videos. The IJB-A evaluation protocol consists of verification (1:1 matching) and identification (1:N search). For verification, each of the 10 splits contains around 11,748 pairs of templates with 1,756 positive and 9,992 negative pairs on average. For identification, the protocol also consists of 10 splits which evaluates the search performance. Examples of IJB-A faces are shown in Figure 2.5. In JANUS CS2, there are about 167 gallery templates and 1763 probe templates. They are used for both identification and verification. The training set for both IJB-A and JANUS CS2 contains 333 subjects, while the test set contains 167 subjects.

Figure 2.5: IJB-A examples. Left 4 are positive pairs and right 4 are negative pairs.

2.3.2 Implementation Details

2.3.2.1 FV-DCNN

For FV-DCNN, each face image is detected and aligned using the open-source library dlib [53, 105]. Faces are aligned into the canonical coordinate using the similarity transform of seven landmark points (*i.e.* two left eye corners, two right eye corners, nose tip, and two mouth corners) and fed into the DCNN network. Training details of the network can be found in [14].

For the LFW dataset, we learn 64 Gaussians with spatial encoding. For the CFP dataset, we learn 64 Gaussians with traditional encoding since the alignment for profile faces is not reliable. A whitening PCA is applied for initializing the joint Bayesian metric learning. A score level fusion is applied on the similarity scores from *pool* and *conv_fv* with a linear weight.

2.3.2.2 VLAD-DCNN

For VLAD-DCNN, each face image is first detected and aligned using the Hyperface method introduced in [80], which is a multi-task DCNN network that can

simultaneously perform face detection, fiducial extraction and gender classification on an input image. Alignment part is the same as FV-DCNN.

The proposed DCNN model is trained on the CASIA-WebFace dataset [121] using caffe [50], without finetuning on the JANUS training set. The data is augmented with horizontally flipped faces. For training, we use 128 as the batch size, set the initial negative slope for PReLU to 0.25, and set the weight decay of all convolutional layers to 0 and of the final fully connected layer to 5e-4. Finally, the learning rate is initially set equal to 1e-2 and reduced by half for every 100,000 iterations. The momentum is set equal to 0.9. The snapshot of 720,000th iteration is used for all our experiments.

For each image, the *conv* features from the training set are normalized after taking the square root (with sign preserved). Two additional dimensions are added as extra spatial information. K-means clustering is then applied on the normalized and augmented features with K=16. The features are encoded using the VLAD technique with $(320 + 2) \times 16 = 5152$ dimensions. After *conv_vlad*, *pool* and *fc* features are extracted, media averaging [23] is applied so that the features coming from the same media (image or video) are averaged.

The training data used for metric learning is the JANUS training set only. Both JB and triplet distance embedding metrics are learned for *conv_vlad* features. Before metric learning, the high dimensional VLAD features are first projected onto a 200-dimensional space by the matrix $\mathbf{P}$ (200×5152) learned using the whitening Principle Component Analysis (WPCA). For JB, the learned matrices $\mathbf{W}$ and $\mathbf{V}$ are both 200×200 ($d = D = 200$). The learning rates γ and γ_β are both set to 1e-2

and the margin α is 1e-3. The proportion between positive pairs and negative pairs in the training set is 1:1. For triplet embedding, the learned projection matrix $\mathbf{W}$ is also 200×200. The learning rate γ and margin α are both 1e-3. Hard negatives are chosen from 100 randomly picked negatives for a given anchor. We call the scores obtained by triplet embedding as **A**, and the scores obtained by the JB metric learning as **B**.

For both *pool* and *fc* features, 128-dimensional triplet embeddings are learned. The learning hyperparameters are the same as **A**. We call the scores obtained from *pool* after triplet embedding as **C** and the scores obtained from *fc* after triplet embedding as **D**. Finally, we fused **A** with **D** and **B** with **C** by linear score level fusion.

2.3.3 Results on the CFP dataset

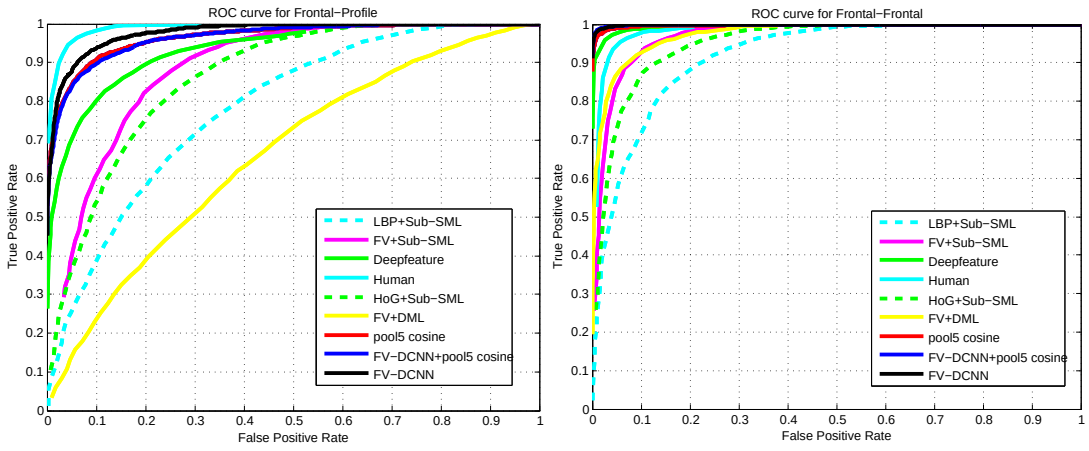
First, to investigate how pose variations influence the performance of the proposed FV-DCNN method, we conduct experiments on CFP.

On this dataset, the human performance for the frontal-to-profile verification is 94.57% accuracy and frontal-to-frontal is 96.24% accuracy. The dataset has been evaluated in [90] using previous state-of-the-art algorithms, including Fisher vector based on SIFT features, Sub-SML [8], and a deep learning approach which uses a similar architecture and ReLU as the activation function without applying data augmentation.

The evaluation results and the ROC curves are shown in Table 2.2 and Fig-

	Frontal-Profile			Frontal-Frontal		
Algorithm	Accuracy	EER	AUC	Accuracy	EER	AUC
HoG+Sub-SML	$77.31 \pm 1.61\%$	$22.20 \pm 1.18\%$	$85.97 \pm 1.03\%$	$88.34 \pm 1.33\%$	$11.45 \pm 1.35\%$	$94.83 \pm 0.80\%$
LBP+Sub-SML	$70.02 \pm 2.14\%$	$29.60 \pm 2.11\%$	$77.98 \pm 1.86\%$	$83.54 \pm 2.40\%$	$16.00 \pm 1.74\%$	$91.70 \pm 1.55\%$
FV+Sub-SML	$80.63 \pm 2.12\%$	$19.28 \pm 1.60\%$	$88.53 \pm 1.58\%$	$91.30 \pm 0.85\%$	$8.85 \pm 0.74\%$	$96.87 \pm 0.39\%$
FV+DML	$58.47 \pm 3.51\%$	$38.54 \pm 1.59\%$	$65.74 \pm 2.02\%$	$91.18 \pm 1.34\%$	$8.62 \pm 1.19\%$	$97.25 \pm 0.60\%$
Deep features	$84.91 \pm 1.82\%$	$14.97 \pm 1.98\%$	$93.00 \pm 1.55\%$	$96.40 \pm 0.69\%$	$3.48 \pm 0.67\%$	$99.43 \pm 0.31\%$
Human	$\mathbf{94.57 \pm 1.10\%}$	$\mathbf{5.02 \pm 1.07\%}$	$\mathbf{98.92 \pm 0.46\%}$	$\mathbf{96.24 \pm 0.67\%}$	$\mathbf{5.34 \pm 1.79\%}$	$\mathbf{98.19 \pm 1.13\%}$
pool cosine	$90.41 \pm 1.16\%$	$9.63 \pm 1.21\%$	$96.53 \pm 0.99\%$	$97.79 \pm 0.38\%$	$2.20 \pm 0.36\%$	$99.73 \pm 0.18\%$
conv_fv + *pool* cosine	$89.83 \pm 1.88\%$	$10.40 \pm 1.85\%$	$96.37 \pm 0.97\%$	$\mathbf{98.67 \pm 0.36\%}$	$\mathbf{1.40 \pm 0.37\%}$	$\mathbf{99.90 \pm 0.09\%}$
conv_fv	$\mathbf{91.97 \pm 1.70\%}$	$\mathbf{8.00 \pm 1.68\%}$	$\mathbf{97.70 \pm 0.82\%}$	$98.41 \pm 0.45\%$	$1.54 \pm 0.43\%$	$99.89 \pm 0.06\%$

Table 2.2: Performance comparison of different methods on the CFP dataset.



(a) (b)

Figure 2.6: The ROC curves corresponding to (a) Frontal-Profile matching and (b) Frontal-Frontal matching on the CFP dataset.

ure 2.6, respectively. From the figure, even though performance drops much in the frontal-to-profile setting, the proposed FV-DCNN approach still performs compara-

ble to the human performance and better than *pool* features and other approaches, including the DCNN baseline. Since FV-DCNN encodes spatial and appearance information contained in *conv* features into the high-dimensional feature vector, it is robust to large pose variations than other approaches. Also notice that by fusing *conv_fv* and *pool*, we improve the performance for frontal-to-frontal setting. But frontal-to-profile setting is not as good as single *conv_fv*. This is because under extreme poses, global features are not robust and will degrade the overall performance.

2.3.4 Results on the LFW dataset

We show the mean accuracy of the proposed FV-DCNN representation with other state-of-the-art deep learning-based methods on the LFW dataset: DeepFace [102], DeepID2 [98], DeepID3 [97], FaceNet [89], Yi *et al.* [121], Wang *et al.* [106], and human performance. The results are summarized in Table 2.3.

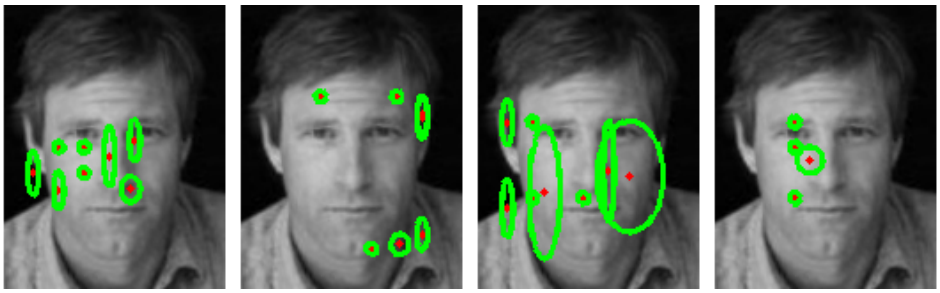
Table 2.3 shows that the proposed FV-DCNN performs comparable to many other deep learning-based methods. In addition, it also shows that the error reduces when we fuse the similarity scores of both *conv_fv* representation (local descriptor) and *pool* representation (global descriptor). Note that some of the deep learning-based methods compared in Table 2.3 use millions of data samples for training the model that typically has tens of millions of parameters or fuse multiple DCNN models together. In contrast, we use only the CASIA dataset which has less than 500K images to train a single DCNN model with about five million parameters.

Method	#Net	Training Set	Metric	Mean Accuracy ± Std
DeepFace [102]	1	4.4 million images of 4,030 subjects, private	cosine	95.92% ± 0.29%
DeepFace	7	4.4 million images of 4,030 subjects, private	unrestricted, SVM	97.35% ± 0.25%
DeepID2 [98]	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% ± 0.15%
DeepID3 [97]	50	202,595 images of 10,117 subjects, private	unrestricted, Joint-Bayes	99.53% ± 0.10%
FaceNet [89]	1	260 million images of 8 million subjects, private	L2	99.63% ± 0.09%
Yi *et al.* [121]	1	494,414 images of 10,575 subjects, public	cosine	96.13% ± 0.30%
Yi *et al.*	1	494,414 images of 10,575 subjects, public	unrestricted, Joint-Bayes	97.73% ± 0.31%
Wang *et al.* [106]	1	494,414 images of 10,575 subjects, public	cosine	96.95% ± 1.02%
Wang *et al.*	7	494,414 images of 10,575 subjects, public	cosine	97.52% ± 0.76%
Wang *et al.*	1	494,414 images of 10,575 subjects, public	unrestricted, Joint-Bayes	97.45% ± 0.99%
Wang *et al.*	7	494,414 images of 10,575 subjects, public	unrestricted, Joint-Bayes	98.23% ± 0.68%
Ding *et al.* [26]	8	471,592 images of 9,000 subjects, public	unrestricted, Joint-Bayes	99.02% ± 0.19%
Human, funneled [106]	N/A	N/A	N/A	99.20%
pool cosine	1	494,414 images of 10,575 subjects, public	cosine	97.82% ± 0.59%
conv_fv	1	494,414 images of 10,575 subjects, public	unrestricted, Joint-Bayes	97.72% ± 0.61%
conv_fv+*pool* cosine	1	494,414 images of 10,575 subjects, public	unrestricted, Joint-Bayes	98.13% ± 0.40%

Table 2.3: Performance comparison of different methods on the LFW dataset dataset.

2.3.5 Visualization of the Learned GMMs by FV-DCNN

Figure 2.7 shows an image in the LFW dataset along with the last two dimensions (which are the spatial coordinates) of the Gaussians learned by FV-DCNN. Gaussians are learned from the original 320-dimensional *conv* features plus two dimensional spatial features without dimension reduction, from the images in LFW Split 1. We only choose Gaussians whose corresponding energy in the learned projection matrices are among the top eight or bottom eight, which implies the discriminative power of these Gaussians. Figures 2.7(a) and 2.7(b) are Gaussians learned after we apply square root and L_2 normalization on the *conv* features. Figures 2.7(c) and 2.7(d) are Gaussians learned without any normalization. From Figures 2.7(a)



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

Figure 2.7: (a) Top eight Gaussians using square root and L_2 normalization. (b) Bottom eight Gaussians using square root and L_2 normalization. (c) Top eight Gaussians without normalization. (d) Bottom eight Gaussians without normalization.

and 2.7(b), the top eight Gaussians are located near eyes, nose and mouth after normalization. The bottom eight Gaussians are out of the face region in general. But in Figures 2.7(c) and (d), without pre-normalization, the top eight Gaussians are

everywhere in the image with large variations in spatial location. Also, the bottom eight Gaussians are all located in the center of the face, which is not expected. This comparison shows that spatial information is not encoded into Gaussians if we do not apply normalization before learning the Gaussians.

2.3.6 Results on IJB-A and JANUS CS2 datasets

We compare VLAD-DCNN with FV-DCNN and other methods on IJB-A and JANUS CS2. For a fair comparison, here we learn a 16-component GMM for FV-DCNN. *conv_fv* are computed from *conv* feature maps after square root normalization and spatial encoding. The encoded FVs are of $(320+2)\times 32 = 10304$ dimensions. Triplet embedding with 200 dimensions is learned with the same hyperparameters as **A**, **C** and **D**. We denote this result by **FV-DCNN**.

We also compare our methods with two recent methods [1] and [65]. The verification and identification results corresponding to different methods on the CS2 and IJB-A datasets are shown in Table 2.4, 2.5 and Figure 2.8.

To clarify the notation again, in the following tables and figures, **A** is *conv_vlad* with triplet embedding. **B** is *conv_vlad* with JB metric. **C** is *pool* with triplet embedding. **D** is *fc* with triplet embedding. **FV-DCNN** is *conv_fv* with triplet embedding. **B+C** and **A+D** correspond to score level fusion.

From the tables and curves we can see that before fusion, **D** has the best IJB-A verification result. The best CS2 at 1e-2 result is achieved by **A**, which implies that VLAD encoding does extract more information from the features maps of the last

FAR	CS2			IJB-A (1:1)		
	1e-3	1e-2	1e-1	1e-3	1e-2	1e-1
[1]	-	89.7%	95.9%	-	78.7%	91.1%
[65]	82.4%	92.6%	-	72.5%	88.6%	-
FV-DCNN	81.83%	91.46%	97.53%	72.94%	86.63%	95.80%
A	82.92%	92.44%	97.71%	73.64%	87.65%	96.16%
B	82.34%	92.14%	97.76%	73.31%	87.11%	96.17%
C	83.42%	91.71%	97.53%	77.09%	88.21%	96.18%
D	84.04%	92.05%	97.52%	**77.88%**	88.70%	96.22%
B+C	84.43%	92.66%	**97.90%**	76.62%	88.70%	96.56%
A+D	**84.69%**	**92.72%**	97.85%	77.36%	**88.85%**	**96.66%**

Table 2.4: CS2 and IJB-A Verification Results.

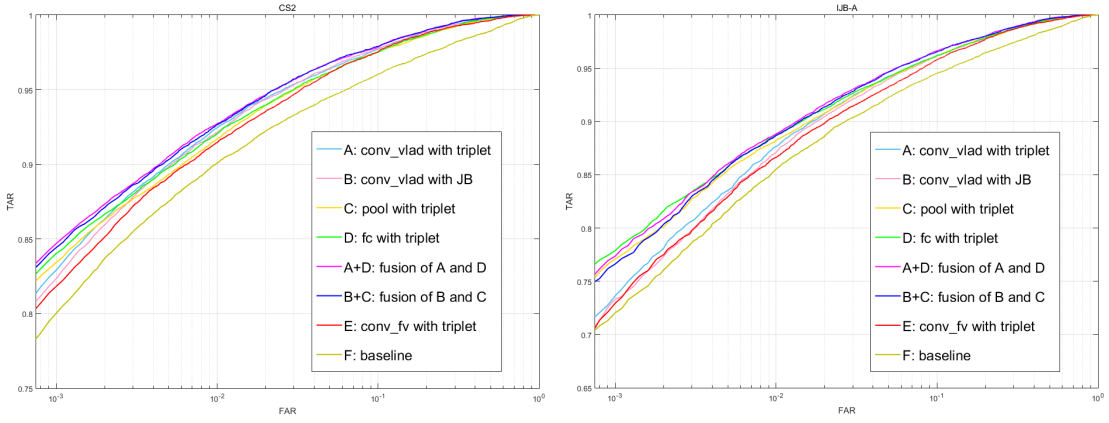


Figure 2.8: Results on the JANUS CS2 and IJB-A datasets. (a) the average ROC curves for the JANUS CS2 verification protocol and (b) the average ROC curves for IJB-A verification protocol over 10 splits.

Accuracy	CS2			IJB-A		
	rank 1	rank 5	rank 10	rank 1	rank 5	rank 10
[1]	86.5%	93.4%	94.9%	84.6%	92.7%	94.7%
[65]	89.8%	**95.6%**	**96.9%**	90.6%	**96.2%**	**97.7%**
FV-DCNN	88.80%	94.60%	96.10%	90.00%	95.20%	96.60%
A	89.20%	94.80%	96.00%	90.40%	95.30%	96.30%
B	89.10%	94.70%	95.90%	90.40%	95.20%	96.20%
C	89.30%	94.60%	95.90%	90.50%	95.20%	96.50%
D	89.70%	94.70%	96.00%	90.90%	95.30%	96.60%
B+C	89.90%	95.00%	96.40%	91.00%	95.70%	96.80%
A+D	**90.20%**	95.10%	96.40%	**91.30%**	95.60%	96.90%

Table 2.5: CS2 and IJB-A Identification Results.

convolutional layer, instead of direct average pooling. Its IJB-A performance is also comparable with **C** and **D**. After fusing **B+C**, CS2 at 1e-2 increases about 0.9% from **C**. IJB-A at 1e-2 also has a 0.5% gain, which shows the effectiveness of our fusion strategy. After fusing **A+D**, we obtain the best results on both CS2 at 1e-2 and IJB-A at 1e-2. Also, **A** performs better than **FV-DCNN** at both CS2 1e-2 and IJB-A 1e-2 with a gap of about 1%. All of the above results show that based on DCNN features in this scenario, VLAD-DCNN is very competitive and performs better than FV-DCNN.

Compared with [1] and [65], our methods performs consistently better for verification task on both CS2 and IJB-A. For identification task at Rank 5 and 10, our performance is slightly lower but still comparable to [65]. It is because in [65] the CASIA WebFace dataset is expanded to over 2.4 Million images for training using 3D synthesized image. But our model is trained using the original CASIA dataset without any augmentation.

2.3.7 Comparison between FV-DCNN and VLAD-DCNN

We observe that FV-DCNN does not perform as well as VLAD-DCNN on IJB-A and CS2. Our explanation is that the second order statistics are not helpful in this scenario. The bag-of-words method is usually designed for low-level local features like SIFT, SURF or HoG, which are basically histograms. In these cases, for a set of histograms of local features, both the first order (mean of the histograms) and the second order (variance of every entry of the histograms) statistics contain

discriminative information. But for DCNN features, the second order statistics are much less important than the first order ones. Different from the traditional local features, the DCNN features extracted from the high level layers are already very discriminative. As discussed in previous section, they are not as local as traditional local features since their receptive fields in the input image are getting bigger as the network is getting deeper. Therefore, the variance of the set of DCNN features from the same image is more likely to contain noise than useful information. When computing FV, we need to scale each entry of the feature according to its variance and aggregate these shifted and scaled features together as

$$\mathbf{\Phi}_{ik}^{(1)} = \frac{1}{N\sqrt{w_k}} \sum_{p=1}^{N} \alpha_k(\mathbf{v}_p) \left(\frac{\mathbf{v}_{ip} - \boldsymbol{\mu}_{ik}}{\boldsymbol{\sigma}_{ik}} \right), \tag{2.16}$$

$$\mathbf{\Phi}_{ik}^{(2)} = \frac{1}{N\sqrt{2w_k}} \sum_{p=1}^{N} \alpha_k(\mathbf{v}_p) \left(\frac{(\mathbf{v}_{ip} - \boldsymbol{\mu}_{ik})^2}{\boldsymbol{\sigma}_{ik}^2} - 1 \right). \tag{2.17}$$

If the variances are not reliable enough, it will degrade the performance.

In contrast, since the DCNN features are robust and discriminative, the first order statistics still contain important information (even more robust after taking the average over the neighborhood). VLAD only considers the first order statistics and will not be affected by the noise variance. Thus compared to FV-DCNN, VLAD-DCNN preserves useful information.

To examine the above assumption, we design another experiment based on the verification protocols of CS2 and IJB-A Split 1. We first learn a GMM of 16 components based on the training set of Split 1. Then we randomly choose one position in the 7×7 feature map of *conv* features. Given an image, instead of average pooling these 320-dimensional local features or performing VLAD encoding

to get the output features, we pick the 320-dimensional local feature at the randomly selected position from the $7\times7\times320$ *conv* features and consider it as a representation of this image. In this way, every image is directly represented by the local features at a certain position in the feature map. Then we encode them in two ways. One follows the VLAD encoding by subtracting the features by their nearest GMM mean as $\mathbf{x}_{vlad} = \mathbf{x} - \mathbf{m}_{nn}$ without encoding the variance information. The other method mimics the FV encoding by subtracting the features by their nearest GMM mean and dividing by the corresponding standard deviation, which is $\mathbf{x}_{fv} = (\mathbf{x} - \mathbf{m}_{nn})/\boldsymbol{\sigma}_{nn}$.

The objective of this experiment is to see whether encoding the second order statistics of the DCNN features will reduce the quality of these local features. Since the FV feature is the aggregation of encoded local features, if the performance of encoded local features decreases, it will very likely affect the performance of FV features. We evaluated the performance of both encoded features with cosine distance. The verification results averaged over 10 trials on CS2 and IJB-A Split 1 are shown in Table 2.6.

FAR	CS2			IJB-A (1:1)		
	1e-3	1e-2	1e-1	1e-3	1e-2	1e-1
$\mathbf{x}_{vlad}$	**50.33%**	**67.77%**	**84.22%**	**41.26%**	**62.26%**	**80.07%**
$\mathbf{x}_{fv}$	49.71%	67.43%	84.18%	40.63%	61.55%	79.81%

Table 2.6: CS2 and IJB-A Verification Results of Encoded Local Features.

From Table 2.6 we see that the VLAD-like encoded local DCNN features per-

form consistently better than FV-like encoded local DCNN features, which supports our assertion that the second order statistics of the DCNN features contain little discriminative information. It also explains why FV-DCNN's performance on IJB-A and CS2 is not as good as VLAD-DCNN.

2.4 Concluding Remarks

In this chapter, we proposed FV-DCNN and VLAD-DCNN for unconstrained face recognition which combines FV/VLAD encoding with DCNN features. We demonstrated the effectiveness of FV-DCNN on LFW and the challenging CFP dataset with large pose variations. It was shown that the FV-DCNN method captures both local and global variations in convolutional features. Experiments on the challenging IJB-A and JANUS CS2 datasets show the effectiveness of VLAD-DCNN. We also compared the performance of VLAD-DCNN and FV-DCNN on IJB-A and JANUS CS2 datasets. We observed that VLAD encoding works better than FV encoding for unconstrained face recognition because the noisy second order statistics used by FV encoding deteriorate its performance.

Chapter 3: An Automatic System for Unconstrained Video-based Face Recognition

3.1 Introduction

Video-based face recognition is an active research topic because of a wide range of applications including visual surveillance, access control, video content analysis, etc. Compared to still face recognition, video-based face recognition is more challenging due to a much larger amount of data to be processed and significant intra/inter-class variations caused by motion blur, low video quality, occlusion, frequent scene changes, and unconstrained acquisition conditions.

To develop the next generation of unconstrained video-based face recognition systems, two datasets have been recently introduced, IARPA Benchmark B (IJB-B) [112] and IARPA Janus Surveillance Video Benchmark (IJB-S) [52], acquired under more challenging scenarios, compared to the Multiple Biometric Grand Challenge (MBGC) dataset [67] and the Face and Ocular Challenge Series (FOCS) dataset [69] which are collected in relatively controlled conditions. IJB-B and IJB-S datasets are captured in unconstrained settings and contain faces with much more intra/inter class variations on pose, illumination, occlusion, video quality, scale, etc.

The IJB-B dataset is a template-based dataset that contains 1845 subjects with 11,754 images, 55,025 frames and 7,011 videos where a template consists of a varying number of still images and video frames from different sources. These images and videos are collected from the Internet and are totally unconstrained, with large variations in pose, illumination, image quality etc. Samples from this dataset are shown in Figure 3.1. In addition, the dataset comes with protocols for 1-to-1 template-based face verification, 1-to-N template-based open-set face identification, and 1-to-N open-set video face identification. For the video face identification protocol, the gallery is a set of still-image templates. The probe is a set of videos (e.g. news videos), each of which contains multiple shots with multiple people and one bounding box annotation to specify the subject of interest. Probes of videos are searched among galleries of still images. Since the videos are composed of multiple shots, it is challenging to detect and associate the faces of the subject of interest across shots due to large appearance changes. In addition, how to efficiently leverage information from multiple frames is another challenge, especially when the frames are noisy.

Similar to the IJB-B dataset, the IJB-S dataset is also an unconstrained video dataset focusing on real world visual surveillance scenarios. It consists of 202 subjects from 1421 images and 398 surveillance videos, with 15,881,408 bounding box annotations. Samples of frames from IJB-S are shown in Figure 3.2. Three open-set identification protocols accompany this dataset for surveillance video-based face recognition where each video in these protocols is captured from a static surveillance camera and contains single or multiple subjects: (1) in surveillance-to-single proto-

Figure 3.1: Example frames of a multiple-shot probe video in the IJB-B dataset. The target annotation is in the red box and face detection results from face detector are in green boxes.

col, probes collected from surveillance videos are searched in galleries consisting of one single high-resolution still image; (2) in surveillance-to-booking protocol, same probes are searched among galleries consisting of seven high-resolution still face images covering frontal and profile poses. Probe templates in (1) and (2) should be detected and constructed by the recognition system itself; (3) in the most challenging surveillance-to-surveillance protocol, both gallery and probe templates are from videos, which implies that probe templates need to be compared with relatively low quality gallery templates.

From these datasets, we summarize the four common challenges in video-based face recognition as follows:

Figure 3.2: Example frames of two single-shot probe videos in the IJB-S dataset.

1. For video-based face recognition, test data are from videos where each video contains tens of thousands of frames and each frame may have several faces. This makes the scalability of video-based face recognition a challenging problem. In order to make the face recognition system to be operationally effective, each component of the system should be fast, especially face detection, which is often the bottleneck in recognition.

2. Since faces are mostly from unconstrained videos, they have significant variations in pose, expression, illumination, blur, occlusion and video quality. Thus, any face representation we design must be robust to these variations and to

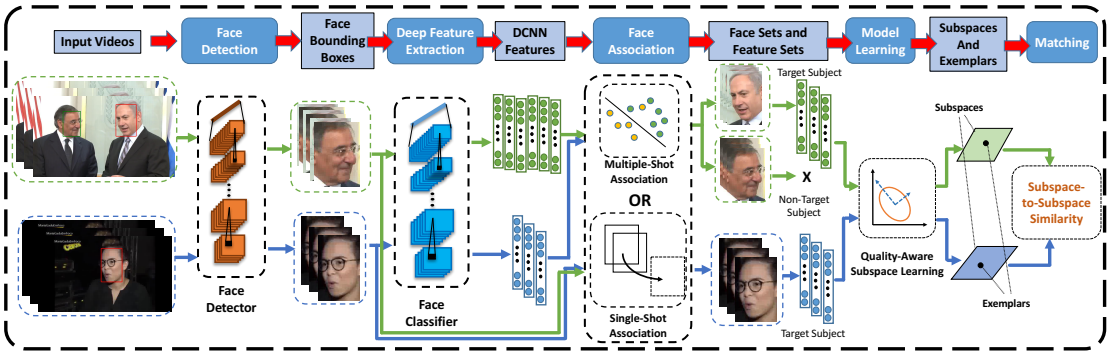


Figure 3.3: Overview of the proposed system.

errors in face detection and association steps.

3. Faces with same identity across different video frames must be grouped by a reliable face association method. Face recognition performance will degrade if faces with different identities are grouped together. Videos in the IJB-B dataset are acquired from multiple shots involving scene and view changes, while most videos in IJB-S are low-quality remote surveillance videos. These conditions increase the difficulty of face association.

4. Since each video contains different number of faces for each identity, the next challenge is how to efficiently aggregate a varying-length set of features from the same identity into a fixed-size or unified representation. Exploiting the correlation information in a set of faces generally results in better performance than using only a single face.

In this chapter, we mainly focus on the second and fourth challenges. After face association, video faces from same identities are associated into sets and the correlation between samples in the same set is leveraged to improve the face recog-

nition performance. For deep representation augmentation methods in video-based face recognition, temporal deep learning model such as Recurrent Neural Network (RNN) can be applied to yield a fixed-size encoded face representation. However, large-scale labeled training data is needed to learn robust representations, which is very expensive to collect in the context of video-based recognition problem. This is also true for the adaptive pooling method [61,119] for image set-based face recognition problem. For IJB-B and IJB-S datasets, lack of large-scale training data makes it impossible to train an RNN-based method. Also, RNN can only work on sequential data, while faces associated from videos are sometimes without a certain order. On the contrary, representative and discriminative models based on manifolds and subspaces have also received attention for image set-based face recognition [107,109]. These methods model sets of image samples as manifolds or subspaces and use appropriate similarity metric for set-based identification and verification. One of the main advantages of subspace-based methods is that different from sample mean, the subspace representation encodes the correlation information between samples. In low-quality videos, faces have significant variations due to blur, extreme poses and low resolution. Exploiting correlation between samples by subspaces will help learn a more robust representation to capture these variations. Also, a fixed-size representation is learned from an arbitrary number of video frames.

To summarize, we propose an automatic system by integrating deep learning components to overcome the challenges in unconstrained video-based face recognition. The proposed system first detects faces and facial landmarks using two state-of-the-art DCNN face detectors, the Single Shot Detector (SSD) for faces [13] and

the Deep Pyramid Single Shot Face Detector (DPSSD) [79]. Next, we extract deep features from the detected faces using state-of-the-art DCNNs [79] for face recognition. SORT [5] and TFA [10] are used for face association in single-shot/multiple-shot videos respectively. Finally, in the proposed face recognition system, we learn a subspace representation from each video template and match pairs of templates using principal angles-based subspace-to-subspace similarity metric on the learned subspace representations. An overview of the proposed system is shown in Figure 3.3.

We evaluate our face recognition system on the challenging IJB-B and IJB-S datasets, as well as MBGC and FOCS datasets. The results demonstrate that the proposed system achieves improved performance over other deep learning-based baselines and state-of-the-art approaches.

3.2 Related Work

1. Deep Learning for Face Recognition: Taigman *et al.* [102] learned a DCNN model on the frontalized faces generated from 3D shape models built from face dataset. Sun *et al.* [96, 98] achieved results surpassing human performance for face verification on the LFW dataset [45]. Schroff *et al.* [89] adopted the GoogLeNet trained for object recognition to face recognition and trained on a large-scale unaligned face dataset. Parkhi *et al.* [72] achieved impressive results using a very deep convolutional network based on VGGNet for face verification. Ding *et al.* [27] proposed a trunk-branch ensemble CNN model for video-based face recognition. Chen

et al. [14] trained a 10-layer CNN on CASIAWebFace dataset [121] followed by the JB metric and achieved state-of-the-art performance on the IJB-A [55] dataset. Chen *et al.* [15] further extended [14] and designed an end-to-end system for unconstrained face recognition and reported very good performance on IJB-A, JANUS CS2, LFW and YouTubeFaces [113] datasets. [17, 124] combined feature encoding with deep neural networks. Bodla *et al.* [6] fused multiple networks to improve face recognition performance. In order to tackle the training bottleneck for face recognition network, Ranjan *et al.* [78] proposed the crystal loss to train the network on very large scale training data. It achieved good performance on IJB-C [66]. Zheng *et al.* [126] achieved good performance on video face datasets including IJB-B [112] and IJB-S [52]. [25] presents a recent face recognizer with state-of-the-art performance.

2. Image Set/Video-based Recognition: For image set-based recognition, Wang *et al.* [109] proposed a Manifold-to-Manifold Distance (MMD) for face recognition based on image set. In [108], the proposed approach models the image set with its second-order statistic for image set classification. Chen *et al.* [20] and [21] proposed a video-based face recognition algorithm using sparse representations and dictionary learning. [125, 127] are recent works on unconstrained video-based face recognition.

3.3 Method

For each video, we first detect faces from video frames and align them using the detected fiducial points. Deep features are then extracted for each detected face

using our DCNN models for face recognition. Based on different scenarios, we use face association or face tracking to construct face templates with unique identities. For videos with multiple shots, we use the face association technique TFA [10] to collect faces from the same identities across shots. For single-shot videos, we use the face tracking algorithm SORT introduced in [5] to generate tracklets of faces. After templates are constructed, in order to aggregate face representations in videos, subspaces are learned using quality-aware principal component analysis. Subspaces along with quality-aware exemplars of templates are used to produce the similarity scores between video pairs by a quality-aware principal angle-based subspace-to-subspace similarity metric. In the following sections, we discuss the proposed video-based face recognition system in detail.

3.3.1 Face/Fiducial Detection

The first step in our face recognition pipeline is to detect faces in images (usually for galleries) and videos. We use two DCNN-based detectors in our pipeline based on different distributions of input.

For regular images and video frames, faces are relatively bigger and with higher resolution. We use SSD trained with the WIDER face dataset as our face detector [13]. For small and remote faces in surveillance videos, we use DPSSD [79] for face detection. DPSSD is fast and capable of detecting tiny faces, which is very suitable for face detection in videos.

After raw face detection bounding boxes are generated using either SSD or

DPSSD detectors, we use All-in-One Face [82] for fiducial localization. It is followed by a seven-point face alignment step based on the similarity transform on all the detected faces.

3.3.2 Deep Feature Representation

After faces are detected and aligned, we use the DCNN models to represent each detected face. The models are state-of-the-art networks with different architectures for face recognition. Different architectures provide different error patterns during testing. After fusing the results from different models, we achieve performance better than a single model. Design details of these networks along with their training details are described in Section 3.4.2.

3.3.3 Face Association

In previous steps, we obtain raw face detection bounding boxes using our detectors. Features for the detected bounding boxes are extracted using face recognition networks. The next important step in our face recognition pipeline is to combine the detected bounding boxes from the same identity to construct templates for good face recognition result.

For single-shot videos, which means the bounding boxes of a certain identity will probably be contiguous, we rely on SORT [5] to build the tracklets for each identity. For multi-shot videos, it is challenging to continue tracking across different scenes. In the proposed system, we use [10] to adaptively update the face

associations through one-shot SVMs.

3.3.4 Model Learning: Deep Subspace Representation

After deep features are extracted for each face template, since each template contains a varying number of faces, these features are further encoded into a fixed-size and unified representation for efficient face recognition.

The simplest representation of a set of samples is the sample mean. However, video templates contain faces with different quality and large variations in illumination, blur and pose. Since average pooling treats all the samples equally, the outliers may deteriorate the discriminative power of the representation. Different from other feature aggregation approaches that require a large amount of extra training data which are not available for datasets like IJB-B and IJB-S, we propose a subspace representation for video face templates.

3.3.4.1 Subspace Learning from Deep Representations

A d-dimensional subspace S can be uniquely defined by an orthonormal basis $\mathbf{P} \in \mathbb{R}^{D \times d}$, where D is the dimensionality of features. Given face features from a video sequence $\mathbf{Y} \in \mathbb{R}^{D \times N}$, where N is the sequence length, $\mathbf{P}$ can be found by optimizing:

$$\underset{\mathbf{P},\mathbf{X}}{\text{minimize}} \ \|\mathbf{Y} - \mathbf{P}\mathbf{X}\|_F^2 \quad s.t. \ \mathbf{P}^T\mathbf{P} = \mathbf{I} \tag{3.1}$$

which is the reconstruction error of features $\mathbf{Y}$ in the subspace S. It is exactly the principal component analysis (PCA) problem and can be easily solved by eigenvalue

decomposition. Let $\mathbf{Y}\mathbf{Y}^T = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ be the eigenvalue decomposition, where $\mathbf{U} = \left[\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_D\right]$ are eigenvectors and $\mathbf{\Lambda} = diag\{\lambda_1, \lambda_2, \ldots, \lambda_D\}$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$ are the corresponding eigenvalues, we have $\mathbf{P} = \left[\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_d\right]$ consisting of the first d basis in $\mathbf{U}$. We use **Sub** to denote this basic subspace learning algorithm (3.1).

3.3.4.2 Quality-Aware Subspace Learning from Deep Representations

In a face template from videos, faces contain large variations in pose, illumination, occlusion, etc. Even in a tracklet, faces have different poses because of head movement, or being occluded in some frames because of the interaction with the environment. When learning the subspace, treating the frames equally is not an optimal solution. In our system, the detection score for each face bounding box provided by the face detector can be used as a good indicator of the face quality, as shown in [78]. Hence, following the quality pooling proposed in [78], we propose quality-aware subspace learning based on detection scores. The learning problem is modified (3.1) as

$$\underset{\mathbf{P},\mathbf{X}}{\text{minimize}} \sum_{i=1}^{N} \tilde{d}_i \|\mathbf{y}_i - \mathbf{P}\mathbf{x}_i\|_2^2 \quad s.t.\ \mathbf{P}^T\mathbf{P} = \mathbf{I} \tag{3.2}$$

where $\tilde{d}_i = softmax(ql_i)$ is the normalized detection score of face i, q is the temperature parameter and

$$l_i = \min(\frac{1}{2}\log\frac{d_i}{1-d_i}, t) \tag{3.3}$$

which is upper bounded by threshold t to avoid extreme values when the detection score is close to 1.

Let $\tilde{\mathbf{Y}} = \left[\sqrt{d_1}\mathbf{y}_1, \cdots, \sqrt{d_N}\mathbf{y}_N\right]$ be the normalized feature set, and the corresponding eigenvalue decomposition be $\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T = \tilde{\mathbf{U}}\tilde{\mathbf{\Lambda}}\tilde{\mathbf{U}}^T$. We have

$$\mathbf{P}_D = \left[\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2, \cdots, \tilde{\mathbf{u}}_d\right] \tag{3.4}$$

which consists of the first d bases in $\tilde{\mathbf{U}}$. The new subspace is therefore learned by treating samples differently according to their quality. This quality-aware learning algorithm is denoted as **QSub**.

3.3.5 Matching: Subspace-to-Subspace Similarity for Videos

After subspace representations are learned for video templates, inspired by manifold-to-manifold distance [109], we measure the similarity between two video templates of faces using a subspace-to-subspace similarity metric. In this part, we first introduce the widely used metric based on principal angles. Then we propose several weighted subspace-to-subspace metrics which take the importance of basis directions into consideration.

3.3.5.1 Principal Angles and Projection Metric

One of the mostly used subspace-to-subspace similarity is based on principal angles. The principal angles $0 \leq \theta_1 \leq \theta_2 \leq \cdots \leq \theta_r \leq \frac{\pi}{2}$ between two linear subspaces S_1 and S_2 can be computed by Singular Value Decomposition (SVD).

Let $\mathbf{P}_1 \in \mathbb{R}^{D\times d_1}$, $\mathbf{P}_2 \in \mathbb{R}^{D\times d_2}$, denoting the orthonormal basis of S_1 and S_2, respectively. The SVD is $\mathbf{P}_1^T\mathbf{P}_2 = \mathbf{Q}_{12}\mathbf{\Lambda}\mathbf{Q}_{21}^T$, where $\mathbf{\Lambda} = diag\{\sigma_1, \sigma_2, \ldots, \sigma_r\}$. $\mathbf{Q}_{12}$ and $\mathbf{Q}_{21}$ are orthonormal matrices. The singular values $\sigma_1, \sigma_2, \ldots, \sigma_r$ are exactly

the cosine of the principal angles as $\cos\theta_k = \sigma_k,\ k = 1, 2, \ldots, r$.

Projection metric [31] is a popular similarity metric based on principal angles:

$$s_{PM}(S_1, S_2) = \sqrt{\frac{1}{r}\sum_{k=1}^{r}\cos^2\theta_k} \tag{3.5}$$

Since $\|\mathbf{P}_1^T\mathbf{P}_2\|_F^2 = \|\mathbf{Q}_{12}\mathbf{\Lambda}\mathbf{Q}_{21}^T\|_F^2 = \|\mathbf{\Lambda}\|_F^2 = \sum_{k=1}^{r}\sigma_k^2 = \sum_{k=1}^{r}\cos^2\theta_k$, we have

$$s_{PM}(S_1, S_2) = s_{PM}(\mathbf{P}_1, \mathbf{P}_2) = \sqrt{\frac{1}{r}\|\mathbf{P}_1^T\mathbf{P}_2\|_F^2} \tag{3.6}$$

and there is no need to explicitly compute the SVD. We use **PM** to denote this similarity metric (3.6).

3.3.5.2 Exemplars and Basic Subspace-to-Subspace Similarity

Existing face recognition systems usually use cosine similarity between exemplars to measure the similarity between templates. The exemplar of a template is defined as its sample mean, as $\mathbf{e} = \frac{1}{L}\sum_{i=1}^{L}\mathbf{y}_i$, where $\mathbf{y}_i$ are samples in the template. Exemplars mainly capture the average and global representation of the template. On the other hand, the projection metric we introduced above measures the similarity between two subspaces, which models the correlation between samples. Hence, in the proposed system, we make use of both of them by fusing their similarity scores as the subspace-to-subspace similarity between two video sequences.

Suppose subspaces $\mathbf{P}_1 \in \mathbb{R}^{D\times d_1}$ and $\mathbf{P}_2 \in \mathbb{R}^{D\times d_2}$ are learned from a pair of video templates $\mathbf{Y}_1 \in \mathbb{R}^{D\times L_1}$ and $\mathbf{Y}_2 \in \mathbb{R}^{D\times L_2}$ in deep features respectively, by either **Sub** or **QSub** methods introduced in Section 3.3.4. Their exemplars are $\mathbf{e}_1 = \frac{1}{L_1}\sum_{i=1}^{L_1}\mathbf{y}_{1i}$ and $\mathbf{e}_2 = \frac{1}{L_2}\sum_{i=1}^{L_2}\mathbf{y}_{2i}$ respectively. Combining the orthonormal

bases and exemplars, the subspace-to-subspace similarity can be computed as:

$$\begin{aligned} s(\mathbf{Y}_1, \mathbf{Y}_2) &= s_{Cos}(\mathbf{Y}_1, \mathbf{Y}_2) + \lambda s_{PM}(\mathbf{P}_1, \mathbf{P}_2) \\ &= \frac{\mathbf{e}_1^T \mathbf{e}_2}{\|\mathbf{e}_1\|_2 \|\mathbf{e}_2\|_2} + \lambda \sqrt{\frac{1}{r} \|\mathbf{P}_1^T \mathbf{P}_2\|_F^2} \end{aligned} \tag{3.7}$$

where $s_{Cos}(\mathbf{Y}_1, \mathbf{Y}_2)$ is the cosine similarity between exemplars, denoted as **Cos**, and $s_{PM}(\mathbf{P}_1, \mathbf{P}_2)$ is computed by (3.6). Since the DCNN features are more robust if we keep their signs, instead of using $s_{Cos}^2(\mathbf{Y}_1, \mathbf{Y}_2)$ as in [109] where the sign information is lost, we use $s_{Cos}(\mathbf{Y}_1, \mathbf{Y}_2)$ in our formulation. Accordingly, we also take the square root of the principal angle term to keep the scale consistent. λ here is a hyperparameter that balances the cosine similarity and principal angle similarity. If $\mathbf{P}_i$'s are learned by **Sub**, we denote the whole similarity metric (including exemplars computing and subspace learning) as **Cos+Sub-PM**. If $\mathbf{P}_i$'s are learned by the proposed **QSub**, we denote the similarity as **Cos+QSub-PM**.

3.3.5.3 Quality-Aware Exemplars

In either **Cos+Sub-PM** or **Cos+QSub-PM** we are still using simple average pooling to compute the exemplars. But as discussed in Section 3.3.4, templates consist of faces of different quality. Treating them equally in pooling will let low-quality faces deteriorate the global representation of the template. Therefore, we propose to use the same normalized detection score as in Section 3.3.4 to compute the quality-aware exemplars by $\mathbf{e}_D = \frac{1}{L} \sum_{i=1}^{L} \tilde{d}_i \mathbf{y}_i$, where $\tilde{d}_i = softmax(ql_i)$ and l_i are computed by (3.3). Then, the cosine similarity between the quality-aware

exemplars is

$$s_{QCos}(\mathbf{Y}_1, \mathbf{Y}_2) = \frac{\mathbf{e}_{D1}^T \mathbf{e}_{D2}}{\|\mathbf{e}_{D1}\|_2 \|\mathbf{e}_{D2}\|_2} \tag{3.8}$$

and we denote it as **QCos**. Using the new cosine similarity, the similarity becomes

$$s(\mathbf{Y}_1, \mathbf{Y}_2) = s_{QCos}(\mathbf{Y}_1, \mathbf{Y}_2) + \lambda s_{PM}(\mathbf{P}_1, \mathbf{P}_2) \tag{3.9}$$

If P_i's are learned by **QSub**, the similarity is further denoted by **QCos+QSub-PM**.

3.3.5.4 Variance-Aware Projection Metric

As previously discussed, the projection metric $S_{PM}(S_1, S_2)$ is the square root of the mean square of principle angles between two subspaces and it treats each basis direction in each subspace equally. But these basis vectors are actually eigenvectors of an eigenvalue decomposition problem. Different basis vectors correspond to different eigenvalues, which represents the variance of data in the corresponding direction. Obviously, those basis directions with larger variances contain more information than those with smaller variances. Therefore, based on the variance of each basis direction, we propose a variance-aware projection metric as:

$$s_{VPM}(\mathbf{P}_1, \mathbf{P}_2) = \sqrt{\frac{1}{r} \|\tilde{\mathbf{P}}_1^T \tilde{\mathbf{P}}_2\|_F^2} \tag{3.10}$$

where

$$\tilde{\mathbf{P}}_i = \frac{1}{tr(\log(\mathbf{\Lambda}_i))} \mathbf{P}_i \log(\mathbf{\Lambda}_i) \tag{3.11}$$

$\mathbf{\Lambda}_i$ is a diagonal matrix whose diagonals are eigenvalues corresponding to eigenvectors in $\mathbf{P}_i$. $\frac{1}{tr(\log(\mathbf{\Lambda}_i))}$ is the normalization factor. We use the logarithm of variance to

weight different basis directions in a subspace. This similarity metric is inspired by the Log-Euclidean distance used for image-set classification in [108]. Empirically, we use $\max(0, \log(\mathbf{\Lambda}_i))$ instead of $\log(\mathbf{\Lambda}_i)$ to avoid negative weights. We use **VPM** to denote this similarity metric (3.10).

3.3.5.5 Quality-Aware Subspace-to-Subspace Similarity

By combining the quality-aware subspace learning, quality-aware exemplars and variance-aware projection metric, we propose the quality-aware subspace-to-subspace similarity between two video templates as:

$$s(\mathbf{Y}_1, \mathbf{Y}_2) = s_{QCos}(\mathbf{Y}_1, \mathbf{Y}_2) + \lambda s_{VPM}(\mathbf{P}_{D1}, \mathbf{P}_{D2}) \tag{3.12}$$

where s_{QCos} is defined in (3.8), $\mathbf{P}_{Di}$'s are learned by (3.4) and s_{VPM} is defined in (3.10). This similarity metric is denoted as **QCos+QSub-VPM**. Comparisons of the proposed similarity metrics and other baselines on several challenging datasets are discussed in Section 3.4.

3.4 Experiments

In this section, we report video-based face recognition results for the proposed system on two challenging video face datasets, IJB-B and IJB-S, and compare with other baseline methods. We also provide results on MBGC, and FOCS datasets, to demonstrate the effectiveness of the proposed system. We introduce the details of datasets, protocols and our training and testing procedures in the following sections.

3.4.1 Datasets

IARPA Janus Benchmark B (IJB-B): IJB-B dataset is an unconstrained face recognition dataset. It contains 1845 subjects with 11,754 images, 55,025 frames and 7,011 multiple-shot videos. IJB-B is a template-based dataset where a template consists of a varying number of still images or video frames from different sources. A template can be either image-only, or video-frame-only, or mixed media template. Sample frames from this dataset are shown in Figure 3.1.

In this work, we only focus on the 1:N video protocol of IJB-B. It is an open set 1:N identification protocol where each given probe is collected from a video and is searched among all gallery faces. Gallery candidates are ranked according to their similarity scores to the probes. Top-K rank accuracy and True Positive Identification Rate (TPIR) over False Positive Identification Rate(FPIR) are used to evaluate the performance. The gallery templates are separated into two splits, G_1 and G_2, all consisting of still images. For each video, we are given the frame index with face bounding box of the first occurrence of the target subject, as shown in Figure 3.1. Based on this anchor, all the faces in that video with the same identity should be collected to construct the probes. The identity of the first occurrence bounding box will be considered as the template identity for evaluation.

IARPA Janus Surveillance Video Benchmark (IJB-S): Similar to IJB-B, the IJB-S dataset is also a template-based, unconstrained video face recognition dataset. It contains faces in two separate domains: high-resolution still images for galleries and low quality, remotely captured surveillance videos for probes. It

consists of 202 subjects from 1421 images and 398 single-shot surveillance videos. The number of subjects is small compared to IJB-B, but it is even more challenging due to the low quality of surveillance videos.

Based on the choices of galleries and probes, we are interested in three different surveillance video-based face recognition protocols: surveillance-to-single protocol, surveillance-to-booking protocol and surveillance-to-surveillance protocol. These are all open set 1:N protocols where each probe is searched among the given galleries. Like IJB-B, the probe templates are collected from videos, but no annotations are provided. Thus raw face detections are grouped to construct templates with the same identities.

Galleries consist of only single frontal high resolution image for surveillance-to-single protocol. Galleries are constructed by both frontal and multiple-pose high resolution images for surveillance-to-booking protocol. For the most challenging surveillance-to-surveillance protocol, galleries are collected from surveillance videos as well, with given bounding boxes. In all three protocols, gallery templates are split into two splits, G_1 and G_2. During evaluation, the detected faces in videos are first matched to the ground truth bounding boxes to find their corresponding identity information. The majority of identities appears in each template will be considered as the identity of the template, and will be used for further identification evaluation. Example frames are shown in Figure 3.2. Notice the remote faces are of very low quality.

Multiple Biometric Grand Challenge (MBGC): The MBGC Version 1 dataset contains 399 walking (frontal face) and 371 activity (profile face) video

sequences from 146 subjects. Figure 3.4 shows some sample frames from different walking and activity videos. In the testing protocol, verification is specified by two sets: target and query. The protocol requires the algorithm to match each target sequence with all query sequences. Three verification experiments are defined: walking-vs-walking (WW), activity-vs-activity (AA) and activity-vs-walking (AW).

(a) MBGC Walking

(b) MBGC Activity

(c) FOCS Walking

(d) FOCS Activity

Figure 3.4: Examples of MBGC and FOCS datasets.

Face and Ocular Challenge Series (FOCS): The video challenge of FOCS is designed for frontal and non-frontal video sequence matching. The FOCS UT Dallas dataset contains 510 walking (frontal face) and 506 activity (non-frontal face) video sequences of 295 subjects with frame size of 720×480 pixels. Like MBGC,

FOCS specifies three verification protocols: walking-vs-walking, activity-vs-walking, and activity-vs-activity. In these experiments, 481 walking videos and 477 activity videos are chosen as query videos. The size of target sets ranges from 109 to 135 video sequences. Sample video frames from this dataset are shown in Figure 3.4.

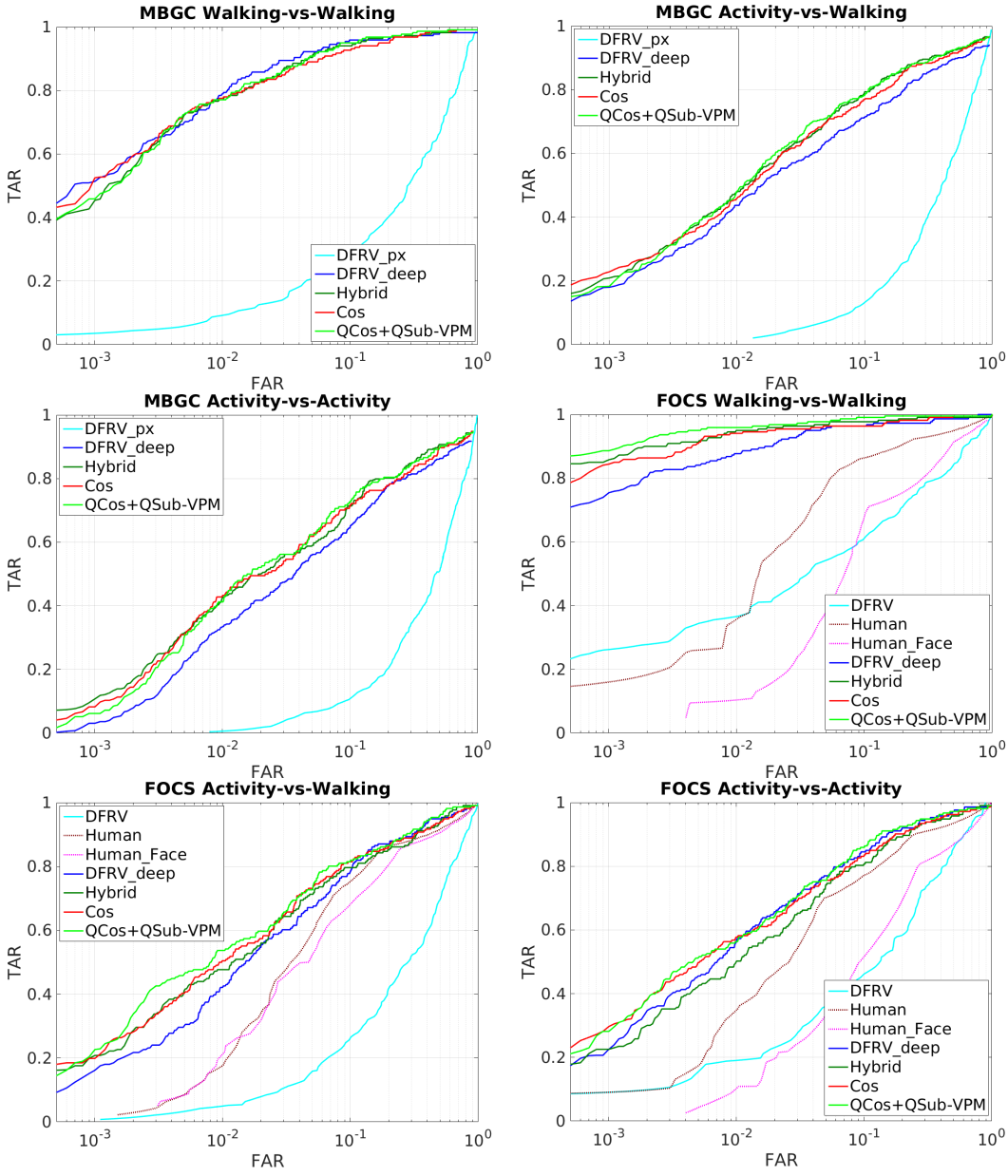


Figure 3.5: Verification results on MBGC and FOCS datasets.

3.4.2 Implementation Details

In this part, we discuss the implementation details for each dataset respectively.

3.4.2.1 IJB-B

For the IJB-B dataset, we employ the SSD face detector [13] to extract the face bounding boxes in all images and video frames. We employ the facial landmark branch of All-in-One Face [82] for fiducial detection on every detected bounding boxes and apply facial alignment based on these fiducials using the seven-point similarity transform.

The aligned faces are further represented using three networks proposed in [81]. We denote them as Network A, Network B and Network C. Network A modifies the ResNet-101 [40] architecture. It has an input size of dimensions 224×224 and adds an extra fully connected layer after the last convolutional layer to reduce the feature dimensionality to 512. Also it replaces the original softmax loss with the crystal loss [78] for more stable training. Network B uses the Inception-ResNet-v2 [99] model as the base network. Similar to Network A, an additional fully-connected layer is added for dimensionality reduction. Naive softmax followed by cross entropy loss is used for this network. Network C is based on the face recognition branch in the All-in-One Face architecture [82]. The branch consists of seven convolutional layers followed by three fully connected layers.

Network A and Network C are trained on the MSCeleb-1M dataset [38] which

contains 3.7 million images from 57,440 subjects. Network B is trained on the union of three datasets called the Universe dataset: 3.7 million still images from the MSCeleb-1M dataset, 300,000 still images from the UMDFaces dataset [4], and about 1.8 million video frames from the UMDFaces Video dataset. For each network, we further reduce its dimensionality into 128 by triplet probabilistic embedding (TPE) [86] trained on the UMDFaces dataset.

For face association, we follows the steps outlined in [10]. Then, features from associated bounding boxes are used to construct the probe templates. We use quality-aware pooling for both gallery and probe templates to calculate their exemplars (**QCos**) where $t = 7$ and $q = 0.3$ are used for detection score normalization. Subspaces are built by applying the quality-aware subspace learning method (**QSub**) on each template and taking the top three eigenvector with the largest corresponding eigenvalues. When fusing the cosine similarity and variance-aware projection similarity metric (**VPM**), we use $\lambda = 1$ so two similarity scores are fused equally. We compute the subspace-to-subspace similarity score for each network independently, and combine the similarity scores from three networks by score-level fusion. We also implement baseline methods using combinations of exemplars from vanilla average pooling (**Cos**), subspaces learned by regular PCA (**Sub**) and projection similarity metric (**PM**).

Methods	Rank=1	Rank=2	Rank=5	Rank=10	Rank=20	Rank=50	FPIR=0.1	FPIR=0.01
[10] with Iteration 0	55.94%	-	68.40%	72.89%	-	83.71%	44.60%	28.73%
[10] with Iteration 3	61.01%	-	73.39%	77.90%	-	87.62%	49.73%	34.11%
[10] with Iteration 5	61.00%	-	73.46%	77.94%	-	87.69%	49.78%	33.93%
Cos	78.37%	81.35%	84.39%	86.29%	88.30%	90.82%	73.15%	52.19%
QCos	78.43%	81.41%	84.40%	86.33%	88.34%	90.88%	**73.19%**	**52.47%**
Cos+Sub-PM	77.99%	81.45%	84.68%	86.75%	88.96%	91.91%	72.31%	38.44%
QCos+Sub-PM	78.02%	81.46%	84.76%	86.72%	88.97%	91.91%	72.38%	38.88%
QCos+QSub-PM	78.04%	81.47%	84.73%	86.72%	88.97%	91.93%	72.39%	38.91%
QCos+QSub-VPM	**78.93%**	**81.99%**	**84.96%**	**87.03%**	**89.24%**	**92.02%**	71.26%	47.35%

Table 3.1: 1:N Search Top-K Average Accuracy and TPIR/FPIR of IJB-B video search protocol.

3.4.2.2 IJB-S

For the IJB-S dataset, we employ the multi-scale face detector DPSSD to detect faces in surveillance videos. We only keep face bounding boxes with detection scores greater than 0.4771, to reduce the number of false detections. We use the facial landmark branch of All-in-One Face [82] as the fiducial detector. Face alignment is performed using the seven-point similarity transform.

Different from IJB-B, since IJB-S does not specify the subject of interest, we are required to localize and associate all the faces for different subjects to yield the probe sets. Since IJB-S videos are single-shot, we use SORT [5] to track every face appearing in the videos. Faces in the same tracklet are grouped to create a probe

template. Since some faces in surveillance videos are of extreme pose, blur and low-resolution, to improve precision, tracklets consisting of such faces should be rejected during the recognition stage. By observation, we find that most of the short tracklets are of low quality and not reliable. The average of the detection score provided by DPSSD is also used as an indicator of the quality of the tracklet. On the other hand, we also want to take the performance of face detection into consideration to strike a balance between recall and precision. Thus in our experiments, we use two configurations for tracklets filtering: 1) We keep those tracklets with length greater than or equal to 25 and average detection score greater than or equal to 0.9 to reject low-quality tracklets and focusing on precision. It is referred to as **with Filtering**. 2) Following the settings in [52], we produce results without any tracklets filtering and focusing on both precision and recall. It is referred to as **without Filtering**.

Because of the remote acquisition scenario and the presence of blurred probes in the IJB-S dataset, we retrain Network A with the same crystal loss but on the Universe dataset used by Network B. We denote it as Network D. We also retrain Network B with the crystal loss [78] on the same training data. We denote it as Network E. As a result of combining high capacity networks and large scale training data, Networks D and E are more powerful than Networks A, B, and C. As before, we reduce feature dimensionality into 128 using the TPE trained on the UMDFaces dataset.

In IJB-S, subspace learning and matching parts are the same as IJB-B except that we combine the similarity score by score-level fusion from Networks D and E. Notice that for the surveillance-to-surveillance protocol, we only use the single

Network D for representation as Network E is ineffective for low-quality gallery faces in this protocol.

3.4.2.3 MBGC and FOCS

For MBGC and FOCS datasets, we use All-in-One Face for both face detection and facial landmark localization. The MBGC and FOCS datasets contain only one subject in a video in general. Hence, for each frame, we directly use the face bounding box with the highest detection score as the target face. Similar to IJB-S, bounding boxes are filtered based on detection scores. From the detected faces, deep features are extracted using Network D. Since MBGC and FOCS datasets do not provide training data, we also use the TPE trained on UMDFaces dataset to reduce feature dimensionality into 128. For MBGC and FOCS, subspace learning and matching parts are the same as for IJB-B and IJB-S.

3.4.3 Evaluation Results

In the following section, we first show some face association results on IJB-B and IJB-S datasets. Then we compare the performance of the proposed face recognition system with several baseline methods. For each dataset, all the baseline methods listed below use deep features extracted from the same network and with the same face detector.

- **Cos:** We compute the cosine similarity scores directly from the exemplars with average pooling.

- **QCos:** We compute the cosine similarity scores from the exemplars with quality-aware average pooling.

- **Cos+Sub-PM:** Subspace-to-subspace similarity is computed by fusing the plain cosine similarity and plain projection metric, and subspaces are learned by plain PCA.

- **QCos+Sub-PM:** Subspace-to-subspace similarity is computed by fusing the quality-aware cosine similarity and plain projection metric, and subspaces are learned by plain PCA.

- **QCos+QSub-PM:** Subspace-to-subspace similarity is computed by fusing the quality-aware cosine similarity and plain projection metric, and subspaces are learned by quality-aware subspace learning.

- **QCos+QSub-VPM:** Subspace-to-subspace similarity is computed by fusing the quality-aware cosine similarity and variance-aware projection metric, and subspaces are learned by quality-aware subspace learning.

IJB-B: Figures 3.6 and 3.7 show some examples of our face association results using TFA in IJB-B dataset. Table 3.1 shows the Top-K Accuracy results for IJB-B video protocol. For this dataset, besides the baselines, our method is compared with original results in [10] corresponding to different iteration numbers. Results shown are the average of two galleries. Notice that our proposed system and [10] use the same face association method, but we have different networks and feature representation techniques.

Figure 3.6: Examples of face association results by TFA on IJB-B. The target annotation is in the red box, and the associated faces of the target subject are in magenta-colored boxes.

IJB-S: Figure 3.8 shows some examples of our face association results using SORT in IJB-S dataset. Tables 3.2, 3.3 and 3.4 show the results for IJB-S surveillance-to-single protocol, surveillance-to-booking protocol and surveillance-to-surveillance protocol respectively. Notice that under the **with Filtering** configuration, we use

Figure 3.7: Associated faces by TFA corresponding to examples in Figure 3.6. Face images are in the order of the confidence of face association.

the regular top-K average accuracy for evaluation. Under the **without Filtering** configuration, we use the End-to-End Retrieval Rate (EERR) metric proposed in [52] for evaluation. For surveillance-to-surveillance protocol, we show results for two different network configurations as well. We also implement state-of-the-art network ArcFace [25] on IJB-S and compare with our method. Results from ArcFace are shown with the prefix **Arc-**.

Two recent works [35, 36] have reported results on the IJB-S dataset. These

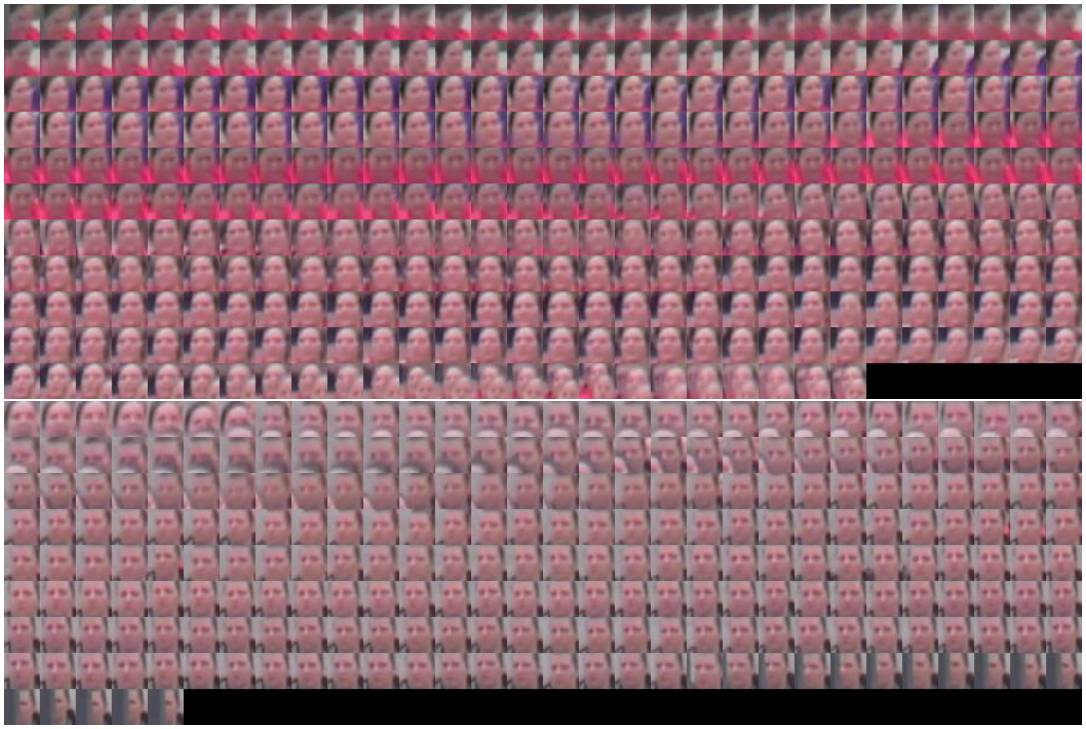


Figure 3.8: Associated faces using SORT in IJB-S. Face images are in their temporal order. Notice the low-quality faces at the boundaries of tracklets since the tracker cannot reliably track anymore.

works mainly focused on face recognition and not detection so that they built video templates by matching their detections with ground truth bounding boxes provided by the protocols and evaluated their methods using identification accuracy and not EERR metric. Our system focuses on detection, association and recognition. Therefore after detection, we associate faces across the video frames to build templates without utilizing any ground truth information and evaluate our system using both identification accuracy and EERR metric. Since these two template building procedures are so different, a directly comparison is not meaningful.

MBGC: The verification results for the MBGC dataset are shown in Table 3.5 and

Methods	Top-K Average Accuracy **with Filtering**						EERR metric **without Filtering**					
	R=1	R=2	R=5	R=10	R=20	R=50	R=1	R=2	R=5	R=10	R=20	R=50
Arc-Cos [25]	52.03%	56.83%	63.16%	69.05%	76.13%	88.95%	24.45%	26.54%	29.35%	32.33%	36.38%	44.81%
Arc-QCos+QSub-PM	60.92%	65.06%	70.45%	75.19%	80.69%	90.29%	28.73%	30.44%	32.98%	35.40%	38.70%	45.46%
Cos	64.86%	70.87%	77.09%	81.53%	86.11%	93.24%	29.62%	32.34%	35.60%	38.36%	41.53%	46.78%
QCos	65.42%	71.34%	77.37%	81.78%	86.25%	93.29%	29.94%	32.60%	35.85%	38.52%	41.70%	46.78%
Cos+Sub-PM	69.52%	75.15%	80.41%	84.14%	87.83%	94.27%	32.22%	34.70%	37.66%	39.91%	42.65%	47.54%
QCos+Sub-PM	69.65%	75.26%	80.43%	84.22%	87.81%	94.25%	32.27%	34.73%	37.66%	39.91%	42.67%	47.54%
QCos+QSub-PM	**69.82%**	**75.38%**	**80.54%**	**84.36%**	**87.91%**	**94.34%**	**32.43%**	**34.89%**	**37.74%**	**40.01%**	**42.77%**	**47.60%**
QCos+QSub-VPM	69.43%	75.24%	80.34%	84.14%	87.86%	94.28%	32.19%	34.75%	37.68%	39.88%	42.56%	47.50%

Table 3.2: 1:N Search results of IJB-S surveillance-to-single protocol. Using both Networks D and E for representation.

Figure 3.5. We compare our method with the baseline algorithms, **Hybrid** [125] and [20] using either raw pixels as $\mathbf{DFRV}_{px}$ (reported in their paper) or deep features as $\mathbf{DFRV}_{deep}$ (our implementation). We also report the results of the proposed method applied on the ArcFace features with the prefix **Arc-**. Figure 3.5 does not include all the baselines, for a clearer view. The result of [20] is not in the table because the authors did not provide exact numbers in their paper.

FOCS: The verification results of FOCS dataset are shown in Table 3.5 and Figure 3.5. O'Toole et al. [70] evaluated the human performance on this dataset. In the figures, **Human** refers to human performance with all bodies of target subjects seen and **Human_Face** refers to performance that only faces of the target subjects are seen. Here besides baseline algorithms and **Hybrid** [125], we also compare our method with [20] in either raw pixels as $\mathbf{DFRV}_{px}$ (reported in their paper) or deep features as $\mathbf{DFRV}_{deep}$ (our implementation). We also report the results using Arc-

Methods	Top-K Average Accuracy **with Filtering**						EERR metric **without Filtering**					
	R=1	R=2	R=5	R=10	R=20	R=50	R=1	R=2	R=5	R=10	R=20	R=50
Arc-Cos [25]	54.59%	59.12%	65.43%	71.05%	77.84%	89.16%	25.38%	27.58%	30.59%	33.42%	37.60%	45.05%
Arc-QCos+QSub-VPM	60.86%	65.36%	71.30%	76.15%	81.63%	90.70%	28.66%	30.64%	33.43%	36.11%	39.57%	45.70%
Cos	66.48%	71.98%	77.80%	82.25%	86.56%	93.41%	30.38%	32.91%	36.15%	38.77%	41.86%	46.79%
QCos	66.94%	72.41%	78.04%	82.37%	86.63%	93.43%	30.66%	33.17%	36.28%	38.84%	41.88%	46.84%
Cos+Sub-PM	69.39%	74.55%	80.06%	83.91%	87.87%	**94.34%**	32.02%	34.42%	37.59%	39.97%	42.64%	**47.58%**
QCos+Sub-PM	69.57%	74.78%	80.06%	83.89%	87.94%	94.33%	32.16%	34.61%	37.62%	39.99%	42.71%	47.57%
QCos+QSub-PM	69.67%	74.85%	80.25%	84.10%	88.04%	94.22%	32.28%	34.77%	37.76%	40.11%	**42.76%**	47.57%
QCos+QSub-VPM	**69.86%**	**75.07%**	**80.36%**	**84.32%**	**88.07%**	94.33%	**32.44%**	**34.93%**	**37.80%**	**40.14%**	42.72%	**47.58%**

Table 3.3: 1:N Search results of IJB-S surveillance-to-booking protocol. Using both Networks D and E for representation.

Face features. Similarly, the results of [20] and human performance are not in the table since they did not provide exact numbers.

3.4.4 Discussions

For the IJB-B dataset, we can see that the proposed system performs consistently better than all the results in [10] and the baseline **Cos** on identification accuracy. For open-set metric TPIR/FPIR, the proposed quality-aware cosine similarity achieves better results, but the proposed subspace similarity metric still performs better than [10] with a large margin. For the IJB-S dataset, we have similar observations: the proposed system with subspace-to-subspace similarity metric performs better than **Cos** on surveillance-to-single and surveillance-to-booking protocols, by relatively large margin. It also achieves better accuracy than **Cos** on the surveillance-to-surveillance protocol. We notice that the fusion of Networks D and

Methods	Top-K Average Accuracy **with Filtering**						EERR metric **without Filtering**					
	R=1	R=2	R=5	R=10	R=20	R=50	R=1	R=2	R=5	R=10	R=20	R=50
Arc-Cos [25]	8.68%	12.58%	18.79%	26.66%	39.22%	68.19%	4.98%	7.17%	10.86%	15.42%	22.34%	37.68%
Arc-QCos+QSub-PM	8.64%	12.57%	18.84%	26.86%	39.78%	**68.21%**	**5.26%**	**7.44%**	**11.31%**	15.90%	**22.68%**	**37.83%**
Cos(D+E)	9.24%	12.51%	19.36%	25.99%	32.95%	52.95%	4.74%	6.62%	10.70%	14.88%	19.29%	30.64%
QCos+QSub-VPM(D+E)	**9.56%**	**13.03%**	19.65%	27.15%	35.39%	56.02%	4.77%	6.78%	10.88%	15.52%	20.51%	32.16%
Cos(D)	8.54%	11.99%	19.60%	28.00%	37.71%	59.44%	4.42%	6.15%	10.84%	15.73%	21.14%	33.21%
QCos(D)	8.62%	12.11%	19.62%	28.14%	37.78%	59.21%	4.46%	6.20%	10.80%	15.81%	21.06%	33.17%
Cos+Sub-PM(D)	8.19%	11.79%	19.56%	28.62%	39.77%	63.15%	4.26%	6.25%	10.79%	16.18%	22.48%	34.82%
QCos+Sub-PM(D)	8.24%	11.82%	19.68%	28.68%	39.68%	62.96%	4.27%	6.25%	10.92%	16.18%	22.39%	34.69%
QCos+QSub-PM(D)	8.33%	11.88%	19.82%	28.65%	39.78%	62.79%	4.33%	6.21%	10.96%	16.19%	22.48%	34.69%
QCos+QSub-VPM(D)	8.66%	12.27%	**19.91%**	**29.03%**	**40.20%**	63.20%	4.30%	6.30%	10.99%	**16.23%**	22.50%	34.76%

Table 3.4: 1:N Search results of IJB-S surveillance-to-surveillance protocol. D stands for only using Network D for representation. D+E stands for using both Networks D and E for representation.

E does not work well on surveillance-to-surveillance protocol, especially at higher rank accuracy. Such observations are consistent under both tracklets filtering configurations and their corresponding metrics: **with Filtering** with Top-K average accuracy and **without Filtering** with the EERR metric. The proposed system also outperforms ArcFace with larger margin in surveillance-to-single and surveillance-to-booking protocols of IJB-S. For MBGC and FOCS datasets, from the tables and plots we can see that in general, the proposed approach performs better than **Cos** baseline, $\mathbf{DFRV}_{deep}$, $\mathbf{DFRV}_{px}$ and **Hybrid**.

Figure 3.9 shows the visualization of two templates in IJB-S dataset in PCA-subspace, which illustrates the advantage of the proposed subspace learning method. In the plot, each dot corresponds to a sample in the template, where x- and y-

Methods	MBGC						FOCS					
	WW		AW		AA		WW		AW		AA	
	FAR=0.01	FAR=0.1	FAR=0.01	FAR=0.1	FAR=0.01	FAR=0.1	FAR=0.01	FAR=0.1	FAR=0.01	FAR=0.1	FAR=0.01	FAR=0.1
Arc-Cos [25]	84.40%	92.20%	53.88%	75.00%	32.47%	66.49%	98.18%	**99.09%**	48.61%	69.44%	48.36%	78.87%
Arc-QCos+QSub-PM	**85.32%**	92.20%	**55.58%**	75.00%	32.99%	64.43%	**98.64%**	**99.09%**	52.31%	74.07%	50.23%	79.81%
DFRV$_{deep}$ [20]	78.90%	**95.87%**	43.69%	71.36%	33.51%	64.95%	87.73%	96.36%	42.13%	78.70%	56.81%	84.51%
Hybrid [125]	77.06%	94.04%	48.06%	**79.37%**	42.53%	71.39%	95.00%	97.73%	47.69%	79.63%	50.23%	80.75%
Cos	77.52%	92.66%	45.87%	76.94%	**43.30%**	71.65%	94.09%	96.36%	50.46%	81.48%	57.75%	83.57%
QCos	77.52%	92.66%	47.57%	76.94%	**43.30%**	71.13%	95.91%	**99.09%**	**53.70%**	80.09%	**58.22%**	83.57%
Cos+Sub-PM	77.98%	94.95%	47.57%	79.13%	41.24%	72.68%	91.82%	97.27%	49.07%	**83.33%**	54.93%	85.45%
QCos+Sub-PM	77.98%	94.95%	48.30%	78.64%	41.75%	**73.71%**	95.91%	98.64%	52.78%	82.87%	55.40%	**85.92%**
QCos+QSub-PM	77.52%	94.95%	48.54%	78.64%	41.75%	73.20%	95.91%	**99.09%**	52.31%	81.02%	55.87%	**85.92%**
QCos+QSub-VPM	77.06%	94.95%	48.06%	78.16%	41.24%	72.68%	95.91%	**99.09%**	**53.70%**	81.94%	56.34%	**85.92%**

Table 3.5: Verification results on MBGC and FOCS datasets.

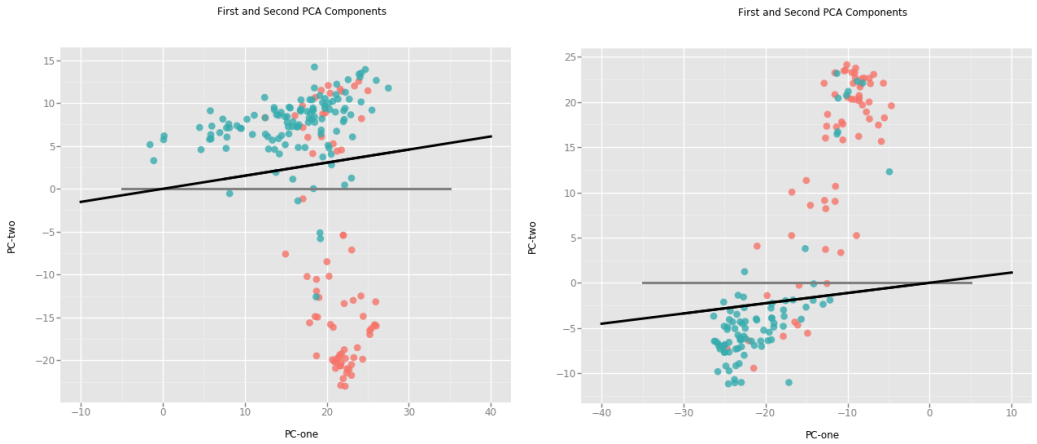


Figure 3.9: Visualization of example templates in IJB-S. Each sample is a dot in the plot with their first two principal components as the coordinates. Samples with $d_i \geq 0.7$ are in **blue** dots and the rest samples are in **red** dots. **Grey** line and **black** line are the projection of the first subspace basis learned by **Sub** and **QSub** respectively.

axes correspond to the first two principal components of the samples, learned from each template respectively. Relatively high-quality detections with detection score greater than or equal to 0.7 are represented by blue dots. Relatively low-quality

detections with detection score less than 0.7 are represented by red dots. The projections of the first subspace bases learned by **Sub** and the proposed **QSub** onto the PCA-subspace are grey and black straight lines in the plot, respectively. From the plot we can see that, with quality-aware subspace learning, the subspaces learned by the proposed method put more weights on the high-quality sample. It fits the high-quality samples better than the low-quality ones. But the plain PCA takes each sample into account equally, which is harmful for the representation of the template.

We also compare our system with other baseline methods as part of an ablation study, from baseline cosine similarity **Cos** to the proposed quality-aware subspace-to-subspace similarity **QCos+QSub-VPM**. As we gradually modify the method by including quality-aware cosine similarity **QCos**, quality-aware subspace learning **QSub** and variance-aware projection metric **VPM**, we can see the performance also gradually improves, especially for IJB-B and IJB-S datasets.

From the results above, we observe the following:

- The proposed system performs the best in general, which shows the effectiveness of 1) learning subspace as template representation, 2) matching video pairs using the subspace-to-subspace similarity metric and 3) utilizing quality and variance information to compute exemplars, learn subspaces and measure similarity.

- **QCos** generally performs better than **Cos**, which shows that quality-aware exemplars weigh the samples according to their quality and better represent

the image sets than plain average exemplars.

- In most of the cases, **Cos+Sub-PM** achieve higher performance than **Cos**. It implies that a subspace can utilize the correlation information between samples and is a good complementary representation of exemplars as global information.

- **QCos+QSub-PM** performs better than **QCos+Sub-PM** in general. It shows that similar to **QCos**, we can learn more representative subspaces based on the quality of samples.

- **QCos+QSub-VPM** works better than **QCos+QSub-PM** in most of the experiments. It implies that by considering the variances of bases in the subspaces, **VPM** similarity is more robust to variations in the image sets.

- The improvement of the proposed system over the compared algorithms is consistent under both **with filtering** and **without filering** configurations on the IJB-S dataset. It shows that our method is effective for both high-quality and low-quality tracklets in surveillance videos.

- For IJB-S, the performance on surveillance-to-surveillance protocol is in general lower than the performance on other protocols. This is because the gallery templates of this protocol are constructed from low-quality surveillance videos, while the remaining two protocols have galleries from high-resolution still images.

- The fusion of Networks D and E does not perform as well as single Network D

on surveillance-to-surveillance protocol, especially at higher rank accuracy. It is probably because of the low-quality galleries in this protocol which Network E cannot represent well.

- On IJB-S, the proposed method performs better than state-of-the-art network ArcFace [25] in general, especially on surveillance-to-single and surveillance-to-booking protocols, which shows the discriminative power of the features from the proposed networks. ArcFace still performs better on surveillance-to-surveillance protocol. But the results also show that using the quality-aware subspace-to-subspace similarity improves the performance for ArcFace features as well.

- On MBGC and FOCS, ArcFace performs better in the walking-vs-walking protocol but Network D outperforms ArcFace on more challenging protocols like activity-vs-activity. Also, by applying the proposed subspace-to-subspace similarity on both features, the performance consistently improves, which shows its effectiveness on different datasets and using different features.

- For the FOCS dataset, the performance of our system surpasses the human performance, which again demonstrates the effectiveness of the proposed system.

3.5 Concluding Remarks

In this chapter, we presented an automatic face recognition system for unconstrained video-based face recognition tasks. The proposed system learns subspaces to represent video faces and matches video pairs by subspace-to-subspace similarity metrics. We evaluated our system on four video datasets and the experimental results demonstrate the superior performance of the proposed system.

Chapter 4: Hybrid Dictionary Learning and Matching for Video-based Face Verification

4.1 Introduction

As we discussed in Chapters 1 and 3, unconstrained video-based face verification is still an open problem due to variations present in video frames including changes in pose, expression, illumination, blurring and low quality of videos. In Chapter 3, we assume that the faces are associated into sets where the temporal orders are ignored. But once the faces are associated by a face tracker into sequences, it is important to also exploit the inherent temporal information available in these sequences.

Over the last decade, generative and discriminative models based on sparse representations have received significant attention in computer vision and pattern recognition [32, 62, 76, 84, 114, 115, 117, 118]. In sparse representation, given samples and a redundant dictionary, the goal is to represent the samples as sparse linear combinations of the dictionary atoms. One of the main advantages of sparse representation-based classification methods is that they are robust to noise. Traditional dictionary learning methods are specifically designed for still images. Lin-

ear Dynamical Systems (LDSs) play an important role in representing sequential data. A wide variety of spatio-temporal signals has been modeled as realizations of LDSs [101]. The idea of sparse representation can be easily incorporated into an LDS model as well. Since sparse representation methods model the video generatively, no pretraining on external data is needed, which is an advantage compared with Long-Short Term Memory (LSTM) [28] and other recurrent neural network-based approaches that require a large-scale labeled training dataset to learn robust representations. Also, based on the observations made in [98], deep features are moderately sparse. This property guarantees that sparse representation is also relevant for deep features.

Classic video-based face recognition algorithms based on sparse representation were presented in [20, 21]. In order to deal with large pose and illumination variations in video sequences, these algorithms cluster the video frames and learn the sparse representation and dictionary for each cluster. The dictionary of the whole video is built by concatenating these dictionaries from different clusters together. The method works well for raw pixels. But in the context of DCNN features, the shortcomings of this method are: 1) Clustering removes the temporal order of video frames. So the dictionaries do not account for temporal correlation. 2) Reconstruction error is used as the similarity metric for recognition tasks. However, DCNN features do not necessarily lie in the Euclidean space because of their high non-linearity. Thus, reconstruction error may not reflect the actual distance between videos.

In this chapter, we propose a hybrid dictionary learning and matching ap-

proach for the unconstrained video-based face verification task, in order to overcome the shortcomings of the method presented in [20,21], and utilize the temporal information in the videos. The proposed method learns both structural and dynamical dictionaries from videos. Structural dictionaries are learned based on the structure of deep representations in videos. Dynamical dictionaries and LDSs are jointly learned using the proposed Linear Dynamical Dictionary Learning (LDDL) algorithm from video sequences. Similar to the method in Chapter 3, with the learned dictionaries, the similarity between videos is measured by subspace-to-subspace similarity instead of the reconstruction error, where the subspaces are spanned by the dictionaries and characterize the local structures of the deep features in videos.

We evaluate our method on MBGC, FOCS and IJB-A datasets to demonstrate that the proposed method performs better than deep learning-based baselines and other state-of-the-art approaches.

4.2 Related Work

Sparse Representation and Dictionary Learning: The K-SVD algorithm [2] is one of the most popular algorithms used for learning sparse representation from data. It learns the dictionary using an optimization algorithm which alternates between sparse coding and dictionary update steps. Besides generative methods, the design of supervised discriminative dictionaries has also received significant attention [51,64,77,120,123]. The advantages of methods in this category are: 1) the dictionaries can be learned from much smaller set of training data than

needed for deep learning approaches, and 2) they can be trained in an unsupervised manner.

Linear Dynamical Systems (LDS): Linear Dynamical Systems have been used to model the evolution of dynamic textured scenes [29, 85]. They offer lower-dimensional representations for videos and have been extensively used for activity modeling, clustering [104] and characterizing the dynamic textures [85]. In [47], each video sequence is modeled as an LDS. Then dictionaries are learned based on the observability matrices of these LDSs. Here the sparsity comes from the generation of observability matrices of LDSs. A sparse coding method based on the LDS model for dynamic textures was proposed in [33], where the LDS was learned from every training video sequence and each testing sequence is modeled as a sparse linear combination of these LDSs. This method is different from the classical dictionary learning method in that sparsity comes from the sparse combination of different dictionaries, not from the combination of atoms in a specific dictionary.

The proposed approach combines the advantages of deep learning, dictionary learning and the LDS model, and is able to learn a compact, robust and discriminative representation for faces in videos for verification.

4.3 Method

An overview of the proposed dictionary learning and matching algorithm for face verification is shown in Figure 4.1. Given a pair of face videos, we first extract their deep features using DCNN models proposed in [14]. For each video, the

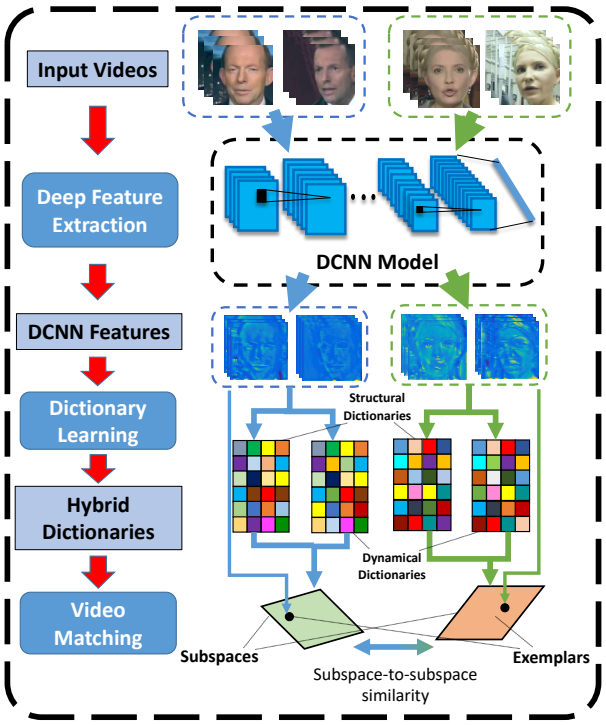


Figure 4.1: Overview of the proposed method.

structural dictionary is learned by solving the basic dictionary learning problem. The dynamical dictionary is learned using the proposed LDDL algorithm. Subspaces spanned by these dictionaries and sample means of the videos (also know as exemplars) are used to produce the similarity scores between the video pair by a subspace-to-subspace similarity metric. Finally, the scores from the structural dictionaries and dynamical dictionaries are fused to produce the final similarity score between two videos.

In the following sections, we discuss the proposed dictionary learning and dictionary-based face matching algorithms in detail.

4.3.1 Dictionary Learning from Deep Features

The performance of a video-to-video matching algorithm depends on how good the learned representation is. The basic problem formulation for video-based face verification is: given faces from the input video sequence $V = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_L\}$, $\mathbf{I}_i \in \mathbb{R}^{C \times C}$, we need to find a robust and discriminative representation for the face appearing in the video. A DCNN model can provide a nonlinear mapping $\phi : \mathbb{R}^{C \times C} \rightarrow \mathcal{S}^M$ that maps the face $\mathbf{I}_i$ into a feature space as $\phi(\mathbf{I}_i)$, which can be more discriminative. Let $\mathbf{Y} = \Big[\phi(\mathbf{I}_1), \cdots, \phi(\mathbf{I}_L)\Big]$ be the sequence of deep representations. Then the problem now reduces to finding a good representation of $\mathbf{Y}$.

In this work, we use sparse dictionary leaning-based approach to augment the deep representations $\mathbf{Y}$ and find a meaningful representation. This can be done in two ways. One way ignores the order of features in $\mathbf{Y}$ and considers them as a set of features. The dictionary learned in this way focuses on characterizing the structure of the feature set. We call it the structural dictionary $\mathbf{D}_s$. The other way treats $\mathbf{Y}$ as an ordered sequence of features and tries to capture the temporal correlation of the features in $\mathbf{Y}$. This is very important for video-based face verification. We call the resulting dictionary as dynamical dictionary $\mathbf{D}_d$. In our approach, we extract meaningful representations from the deep representations by learning both dynamical and structural dictionaries.

Given a video feature sequence $\mathbf{Y} \in \mathbb{R}^{M \times L}$, the Structural Dictionary Learning (SDL) problem is to learn the structure of $\mathbf{Y}$ by solving the following optimization

problem

$$\min_{\mathbf{D}_s,\mathbf{X}} \|\mathbf{Y}-\mathbf{D}_s\mathbf{X}\|_F^2 \ \text{ s.t } \ \|\mathbf{x}_i\|_0 \leq T, \ \ \mathbf{d}_i^T\mathbf{d}_i = 1 \quad \forall i, \tag{4.1}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\|\mathbf{x}\|_0$ is the ℓ_0 norm of $\mathbf{x}$ which counts the number of nonzero elements in $\mathbf{x}$, $\mathbf{D}_s = \Big[\mathbf{d}_1, \ldots, \mathbf{d}_S\Big] \in \mathbb{R}^{M\times S}$ is the structural dictionary of the video, $\mathbf{X} = \Big[\mathbf{x}_1, \ldots, \mathbf{x}_N\Big] \in \mathbb{R}^{S\times L}$ are the sparse coefficients, S is the number of atoms in the dictionary and T is a sparsity parameter. The dictionary $\mathbf{D}_s$ is learned such that the columns of $\mathbf{Y}$ are best represented by the sparse linear combination of atoms in $\mathbf{D}_s$. This problem is the classical dictionary learning problem and can be solved by the K-SVD algorithm [2].

4.3.2 Linear Dynamical Dictionary Learning

In order to learn the dynamical dictionary $\mathbf{D}_d$ that captures the temporal information from the video, we introduce LDS into the dictionary learning framework to model the temporal correlation between frames in a video sequence. The LDS for the sequence can be defined as:

$$\begin{aligned} \mathbf{y}_t &= \mathbf{B}\mathbf{z}_t + \mathbf{w}_t \\ \mathbf{z}_{t+1} &= \mathbf{A}\mathbf{z}_t + \mathbf{v}_t, \end{aligned} \tag{4.2}$$

where $\mathbf{y}_t$ is the observed feature, $\mathbf{z}_t \in \mathbb{R}^S$ is the hidden state of the LDS model, $\mathbf{A} \in \mathbb{R}^{S\times S}$ is the transition matrix, $\mathbf{B} \in \mathbb{R}^{M\times S}$ is the observation matrix. Here, $\mathbf{w}_t \in \mathbb{R}^M$ and $\mathbf{v}_t \in \mathbb{R}^S$ are measurement and process noise, respectively. In this model, the transition matrix $\mathbf{A}$ is introduced to model the linear relationship between the

states of adjacent samples and essentially encodes the temporal information between samples.

If we consider the dynamical dictionary $\mathbf{D}_d$ as the observation matrix $\mathbf{B}$ in the LDS model and combine the basic dictionary learning problem and (4.2) together, the new model would inherit the advantages of both LDS and dictionary learning models. Though in video-based face verification task, the detected faces of the target subject across the entire video contain complex motions from facial expression, head movements, and errors introduced by the face detector and tracker which cannot be modeled by a single linear model. We approach this problem by assuming that after splitting the video into blocks and each with a relatively short length, the motions of faces can be regarded as piece-wise linear. Within each video block, the face motion and detection error motion can be considered to be stationary. Thus, suppose we are given a DCNN feature sequence $\mathbf{Y}$, it is first partitioned into N blocks uniformly so that each block corresponds to a local temporal correlation in the video, as $\mathbf{Y} = [\mathbf{Y}^1, \cdots, \mathbf{Y}^N] \in \mathbb{R}^{M \times \sum L_n}$.

After partitioning the video, the proposed video-specific dictionary learning approach is defined as follows

$$\begin{aligned} \min_{\mathbf{D}_d, \mathbf{X}, \mathbf{A}} & \sum_{n=1}^{N} \|\mathbf{Y}^n - \mathbf{D}_d \mathbf{X}^n\|_F^2 + \eta \sum_{n=1}^{N} \|\mathbf{X}_1^n - \mathbf{A}^n \mathbf{X}_0^n\|_F^2 + \gamma \sum_{n=1}^{N} \|\mathbf{A}^n\|_F^2 \\ \text{s.t.} \quad & \|\mathbf{x}_i^n\|_0 \leq T, \quad \mathbf{d}_i{}^T \mathbf{d}_i = 1 \quad \forall i, n \end{aligned} \tag{4.3}$$

where $\mathbf{d}_i$ is the ith column of $\mathbf{D}_d \in \mathbb{R}^{M \times S}$, which is the overall dynamical dictionary. $\mathbf{X}^n \in \mathbb{R}^{S \times L_n}$ is the sparse coefficients for each partition. $\mathbf{X}_0^n \in \mathbb{R}^{S \times (L_n - 1)}$ and $\mathbf{X}_1^n \in \mathbb{R}^{S \times (L_n - 1)}$ contains the first and last $L_n - 1$ columns of $\mathbf{X}^n$. $\mathbf{A}^n \in \mathbb{R}^{S \times S}$ is the

video specific transition matrix.

Simultaneously solving for $\mathbf{D}_d$, $\mathbf{X}^n$ and $\mathbf{A}^n$ is intractable. Instead, we introduce an auxiliary matrix $\mathbf{W}^n \in \mathbb{R}^{S \times L_n}$ into the optimization problem and solve

$$\begin{aligned} \min_{\mathbf{D}_d, \mathbf{X}, \mathbf{A}, \mathbf{W}} & \sum_{n=1}^{N} \|\mathbf{Y}^n - \mathbf{D}_d \mathbf{X}^n\|_F^2 + \beta \sum_{n=1}^{N} \|\mathbf{X}^n - \mathbf{W}^n\|_F^2 \\ & + \eta \sum_{n=1}^{N} \|\mathbf{W}_1^n - \mathbf{A}^n \mathbf{W}_0^n\|_F^2 + \gamma \sum_{n=1}^{N} \|\mathbf{A}^n\|_F^2 \\ \text{s.t.} \quad & \|\mathbf{x}_i^n\|_0 \leq T \quad , \mathbf{d}_i^T \mathbf{d}_i = 1 \quad \forall i, n \end{aligned} \tag{4.4}$$

where $\mathbf{x}_i^n$ is the ith column of $\mathbf{X}^n$, $\mathbf{W}^n$ is the auxiliary matrix with the same dimension as $\mathbf{X}^n$, $\mathbf{W}_0^n$ and $\mathbf{W}_1^n$ contain the first and last $L_n - 1$ columns of $\mathbf{W}^n$, respectively.

The idea of introducing these auxiliary matrices is to separate the LDS term from the sparse coding term in order to make the optimization more tractable. We use the continuation parameter β to link the sparse coefficients $\mathbf{X}^n$ with the LDS state coefficients $\mathbf{W}^n$. The parameter β is increased in each iteration until it strongly clamps $\mathbf{X}^n$ to $\mathbf{W}^n$. Similar methods have been used previously [122]. Note that in our formulation, we learn a single subject-specific dynamical dictionary $\mathbf{D}_d$. However, the transition matrices $\mathbf{A}^n$ are learned separately from each video block.

4.3.3 Optimization of LDDL

Here we propose an iterative algorithm to solve the optimization problem in (4.4). After introducing auxiliary matrices, we solve for $\mathbf{D}_d$, $\mathbf{X}^n$, $\mathbf{W}^n$ and $\mathbf{A}^n$ iteratively by optimizing with respect to only one variable and fixing the others.

We iterate these steps until the algorithm converges:

4.3.3.1 Solving for $\mathbf{X}$

In this step, we fix $\mathbf{D}_d$ and $\mathbf{W}^n$ and solve for $\mathbf{X}^n$. Note that given $\mathbf{D}_d$, the sparse coefficients $\mathbf{X}^n$ of different video blocks are independent. Thus we can solve for $\mathbf{X}^n$ independently. For each block, it turns into the following optimization problem

$$\begin{aligned} \min_{\mathbf{X}^n} \ & \|(\mathbf{K}^{-1})^T(\mathbf{D}_d^T\mathbf{Y}^n + \beta\mathbf{W}^n) - \mathbf{K}\mathbf{X}^n\|_F^2 \\ \text{s.t.} \ & \|\mathbf{x}_i^n\|_0 \leq T \ \ \forall i \end{aligned} \tag{4.5}$$

where $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ is the eigenvalue decomposition of $\mathbf{D}_d^T\mathbf{D}_d + \beta\mathbf{I}$, $\mathbf{K} = \mathbf{\Lambda}^{\frac{1}{2}}\mathbf{U}^T$, which can be efficiently solved using the Orthogonal Matching Pursuit (OMP) algorithm [74].

4.3.3.2 Solving for $\mathbf{W}$

When $\mathbf{D}_d, \mathbf{X}^n$ and $\mathbf{A}^n$ are fixed, the update for $\mathbf{W}^n$ is obtained by solving the following linear system of equations

$$\mathbf{R}\tilde{\mathbf{w}} = \beta\tilde{\mathbf{x}}. \tag{4.6}$$

where $\tilde{\mathbf{x}} = \left[\mathbf{x}_1^T, \ldots, \mathbf{x}_{L_n}^T\right]^T \in \mathbb{R}^{L_nS}$ is the vectorized version of $\mathbf{X}^n$, $\mathbf{R} = \beta\mathbf{I} + \eta\tilde{\mathbf{A}}_2^T\tilde{\mathbf{A}}_2$, $\tilde{\mathbf{A}}_2 = \mathbf{I} - \tilde{\mathbf{A}}$ and

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{I} & & & \\ \mathbf{A}^n & & & \\ & \ddots & & \\ & & \mathbf{A}^n & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{L_nS \times L_nS}$$

is the reformulated version of $\mathbf{A}^n$. Note that $\mathbf{R}$ is positive definite. Hence, this equation can be solved efficiently by conjugate gradient methods [43]. After obtaining the solution of (4.6), we simply reshape it into $\mathbf{W}^n$.

4.3.3.3 Solving for $\mathbf{A}$

When $\mathbf{W}^n$ is fixed, we obtain the update step for $\mathbf{A}^n$ by an analytical solution

$$\mathbf{A}^n = \mathbf{W}_1^n \mathbf{W}_0^{nT} \left(\mathbf{W}_0^n \mathbf{W}_0^{nT} + \frac{\gamma}{\eta} \mathbf{I} \right)^{-1}. \tag{4.7}$$

4.3.3.4 Solving for $\mathbf{D}_d$

Different from $\mathbf{X}^n$, $\mathbf{W}^n$ and $\mathbf{A}^n$ which are unique for each video block, $\mathbf{D}_d$ is shared by the entire video. When $\mathbf{X}^n$ and $\mathbf{A}^n$ are given, following the atom update procedure in [2], we update the entire dictionary $\mathbf{D}_d$ column by column—updating one atom $\mathbf{d}_i$ at a time as

$$\mathbf{d}_i = \frac{\tilde{\mathbf{Y}} \boldsymbol{\alpha}_i}{\| \tilde{\mathbf{Y}} \boldsymbol{\alpha}_i \|_2}. \tag{4.8}$$

until it converges, where $\tilde{\mathbf{Y}} = \mathbf{Y} - \sum_{j \neq i} \mathbf{d}_j \boldsymbol{\alpha}_j^T$, $\boldsymbol{\alpha}_j^T$ is the jth row of $\mathbf{X}$ and $\mathbf{X} = [\mathbf{X}^1, \cdots, \mathbf{X}^N] \in \mathbb{R}^{S \times \sum L_n}$.

The entire LDDL algorithm is summarized in Algorithm 1.

4.3.4 Dictionary-Based Similarity Metric

After representing video sequences as dictionaries, traditional sparse representation-based methods usually use the reconstruction error as the similarity metric for recognition tasks as in [20,21]. As we mentioned above, DCNN features do not necessarily

Algorithm 1: LDDL algorithm.

Data: Video Sequence:

$\mathbf{Y} = [\mathbf{Y}^1, \cdots, \mathbf{Y}^N]$.

Initialize $\mathbf{D}_d$ using random samples;

for $n = 1 : N$ **do**

Initialize $\mathbf{X}^n$ using OMP [74], initialize $\mathbf{W}^n$ by zeros, initialize $\mathbf{A}^n$ based on initialized $\mathbf{X}^n$;

end

repeat

for $n = 1 : N$ **do**

Update $\mathbf{X}^n$ by solving (4.5), update $\mathbf{W}^n$ by solving (4.6), update $\mathbf{A}^n$ by (4.7) ;

end

Update $\mathbf{D}_d$ by (4.8);

$Iter = Iter + 1$;

until $Convergence$;

Result: Dynamical Dictionary $\mathbf{D}_d$, transition matrices $\{\mathbf{A}^n\}$, sparse coefficients $\{\mathbf{X}^n\}$.

lie in the Euclidean space because of the nonlinearity introduced by activation functions (*e.g.,* sigmoid, tanh, ReLU, etc). Therefore, the reconstruction error is not a good measurement for video sequences in deep features.

Similar to the method introduced in the previous chapter, we model each video using the orthogonal subspace spanned by the learned dictionary of the video.

Even though the features are highly nonlinear, since these local structures on the manifold can be considered as Euclidean by approximation, subspaces can still model the videos properly.

Suppose dictionaries $\mathbf{D}_1 \in \mathbb{R}^{M \times S_1}$ and $\mathbf{D}_2 \in \mathbb{R}^{M \times S_2}$ are learned from a pair of videos $\mathbf{Y}_1 \in \mathbb{R}^{M \times L_1}$ and $\mathbf{Y}_2 \in \mathbb{R}^{M \times L_2}$ in deep feature sequences, we compute two orthonormal bases $\mathbf{P}_1 \in \mathbb{R}^{M \times S_1}$ and $\mathbf{P}_2 \in \mathbb{R}^{M \times S_2}$ corresponding to the dictionaries using the QR decomposition. We also denote their sample means (i.e. exemplar) as $\mathbf{e}_1 = \frac{1}{L_1} \sum_{i=1}^{L_1} \mathbf{y}_{1i}$ and $\mathbf{e}_2 = \frac{1}{L_2} \sum_{i=1}^{L_2} \mathbf{y}_{2i}$. With the orthonormal bases and exemplars, the subspace-to-subspace similarity can be computed as:

$$\begin{aligned} s_M(\mathbf{P}_1, \mathbf{P}_2) &= \cos\theta_0 + \sqrt{\frac{1}{r} \sum_{k=1}^{r} \cos^2 \theta_k} \\ &= \frac{\mathbf{e}_1^T \mathbf{e}_2}{\|\mathbf{e}_1\|_2 \|\mathbf{e}_2\|_2} + \sqrt{\frac{1}{r} \|\mathbf{P}_1^T \mathbf{P}_2\|_F^2} \end{aligned} \tag{4.9}$$

where $\cos\theta_0$ is the cosine similarity between exemplars and $\{\theta_k\}_{i=1}^{r}$ are the principal angles which are the minimal angles between any two basis vectors of the subspaces. Similar to Chapter 4, instead of using $\cos^2\theta_0$ as in [109] where the sign information is lost, we use $\cos\theta_0$ in our formulation. Accordingly, we also take the square root of the principal angle term to keep the scale consistent.

4.3.5 Fusion

After we learn the structural dictionaries $\{\mathbf{D}_{si}\}$ and dynamical dictionaries $\{\mathbf{D}_{di}\}$ as well as subspaces $\{\mathbf{P}_{si}\}$ and $\{\mathbf{P}_{di}\}$, from the videos $\mathbf{Y}_1$ and $\mathbf{Y}_2$, respectively, the overall video-to-video similarity is computed by the weighted sum of the subspace-to-subspace similarity between the structural and dynamical dictionary

pairs respectively as

$$s(\mathbf{Y}_1, \mathbf{Y}_2) = \lambda_1 s_M(\mathbf{P}_{s1}, \mathbf{P}_{s2}) + \lambda_2 s_M(\mathbf{P}_{d1}, \mathbf{P}_{d2}). \tag{4.10}$$

In the experiments, we empirically set $\lambda_1 = \lambda_2 = 0.5$.

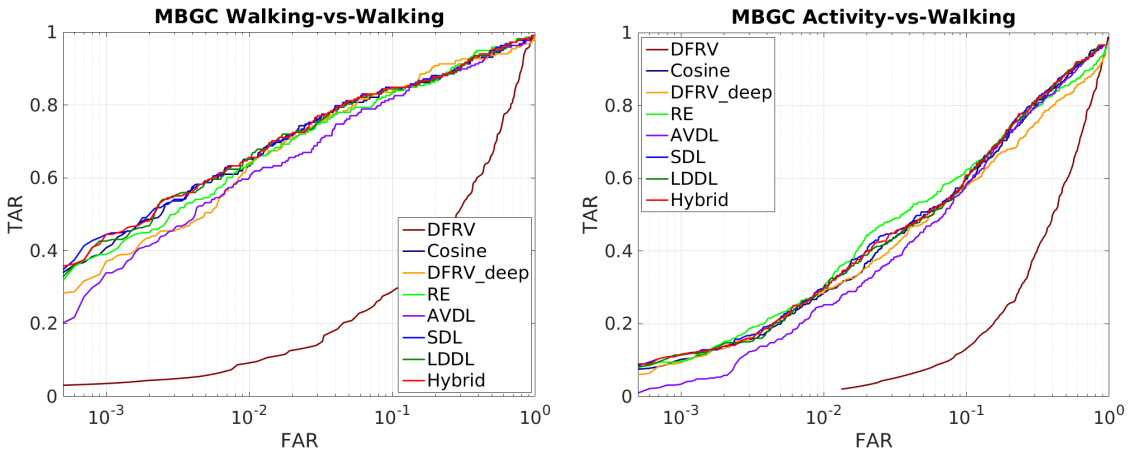
4.4 Experiments

In this section, we present the video-based face verification results of the proposed method using three challenging datasets: MBGC, FOCS and IJB-A.

4.4.1 Implementation Details

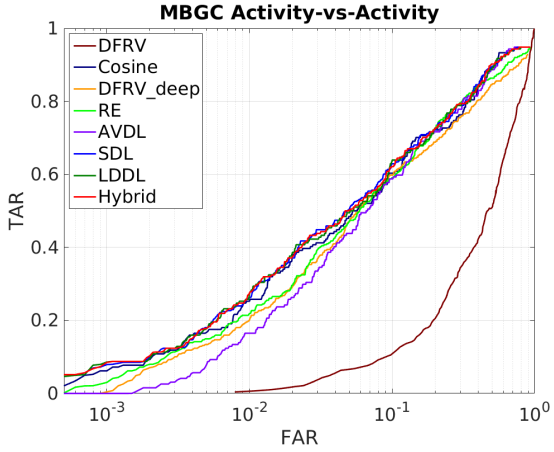
We employ [82] for face detection and facial landmark detection. The MBGC and FOCS datasets contain only a single person in a video. Hence, we directly use the face detection results in our method without association. For the IJB-A datasets, because there are multiple people appearing in a video, we compare the detected face bounding boxes with the ground truth for further improving the detections.

For all datasets, deep representations are first extracted using the DCNN architecture presented in [14]. TPE, a metric learning method proposed in [86], is used to learn an embedding from our external training data and reduce the dimensionality of testing features to 128. Since MBGC and FOCS datasets do not provide training data, we use a subset of external training dataset [4] with 167,877 face images of 3,605 unique subjects. We did not use the training data provided by the IJB-A protocol. To compare with state-of-the-art methods, we also apply our method on the features from the ResNet-101 network trained by the crystal loss introduced in [79].



(a) Walking-vs-Walking

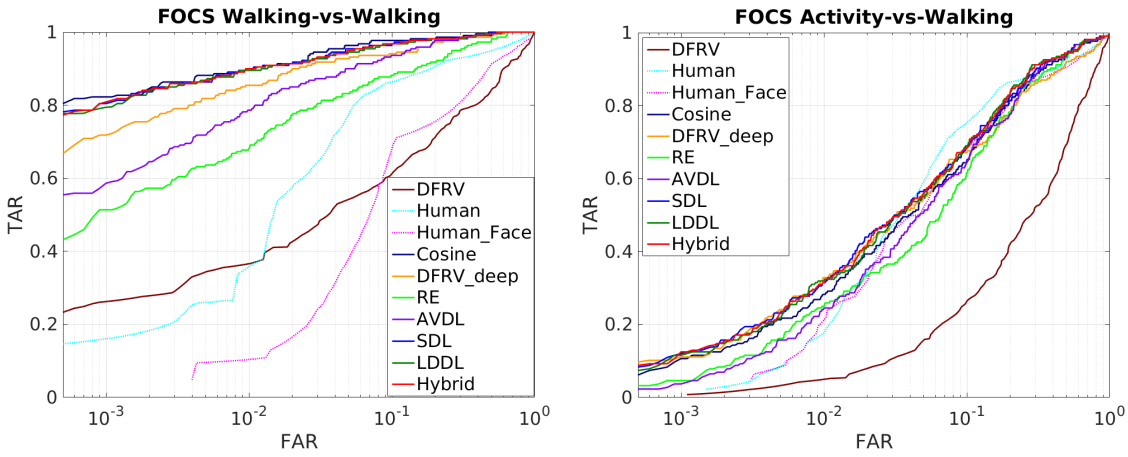
(b) Activity-vs-Walking



(c) Activity-vs-Activity

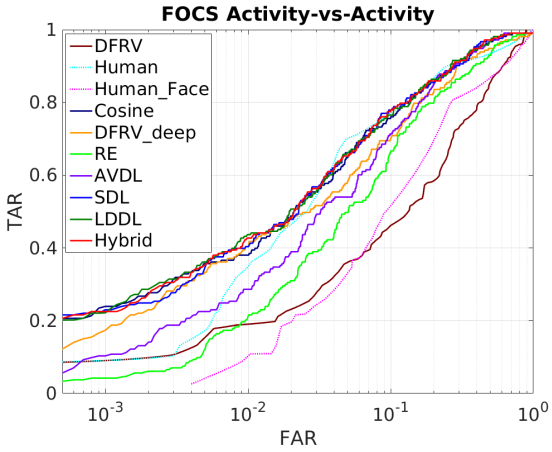
Figure 4.2: Verification results for the MBGC dataset

Given a video (for MBGC and FOCS datasets) or a template containing both still images and video frames (for IJB-A dataset), we learn the dictionaries using the proposed SDL and LDDL algorithms. The corresponding subspaces are computed by QR decomposition from the dictionaries. The similarity scores between video or template pairs are computed using the subspace-to-subspace similarity metric. For template-based IJB-A dataset, we modify the LDDL algorithm so that the still images are also considered when the dynamical dictionaries are updated. The temporal



(a) Walking-vs-Walking

(b) Activity-vs-Walking



(c) Activity-vs-Activity

Figure 4.3: Verification results for the FOCS dataset

correlation constraints are enforced only for video frames.

As we discussed in Section 4.3, to enable the LDS to properly model video sub-segments and to strike a balance between speed and accuracy, we split the videos uniformly into smaller fixed length blocks in with motions are consistent. In MBGC and FOCS datasets, since there are no scene changes in a single video, we uniformly split the videos and empirically fix the block length to be 10 frames. Low quality frames are filtered out based on face detection scores. In IJB-A dataset, the block

length is five frames, which reduces the influence of scene changes in a block. For LDDL, the dictionaries are shared over all blocks from the same video while the transition matrices from different blocks are independent.

In the following section, we compare the performance of the proposed method along with several baseline methods and other dataset specific approaches. All the following baseline methods use deep representations extracted from the same network and with the same face detector, unless otherwise specified.

- **Cosine:** We compute the similarity scores directly using the cosine similarity between sample means of deep representations of video faces.

- **DFRV$_{deep}$:** We implement the adapted version of the video-based face recognition method [20], which uses deep representations instead of pixel intensity.

- **RE:** We compute the similarity scores using the reconstruction error with the learned structural dictionary, to compare with the subspace-to-subspace similarity.

- **AVDL:** We compute the similarity scores using the subspace-to-subspace similarity metric with dynamical dictionary learned by the temporal model based method proposed in [111], to compare with the proposed LDDL algorithm.

- **SDL:** We compute the similarity scores using the subspace-to-subspace similarity metric with the structural dictionary.

- **LDDL:** We compute the similarity scores using the subspace-to-subspace similarity metric with the dynamical dictionary learned by the proposed LDDL

Methods	TAR@FAR on MBGC						TAR@FAR on FOCS					
	WW		AW		AA		WW		AW		AA	
	1%	10%	1%	10%	1%	10%	1%	10%	1%	10%	1%	10%
Cosine	63.30%	83.49%	28.64%	58.50%	25.26%	60.31%	90.00%	97.72%	28.24%	64.81%	38.03%	75.59%
$DFRV_{deep}$	64.22%	83.95%	28.64%	58.01%	20.10%	59.79%	85.45%	94.09%	32.87%	67.13%	40.38%	69.95%
RE	63.76%	83.49%	30.58%	62.38%	21.91%	59.02%	68.18%	88.18%	25.00%	61.57%	21.13%	66.20%
AVDL	60.09%	81.65%	25.24%	58.25%	16.49%	58.76%	78.64%	93.64%	24.54%	64.35%	28.64%	71.36%
SDL	65.14%	84.86%	29.85%	59.71%	26.29%	62.89%	89.55%	96.36%	31.48%	68.98%	40.38%	77.93%
LDDL	65.14%	84.40%	29.61%	60.44%	27.32%	63.92%	89.55%	96.82%	31.94%	68.98%	43.66%	76.53%
Hybrid	65.60%	84.86%	29.61%	60.92%	27.58%	62.11%	90.00%	96.82%	31.48%	68.52%	42.72%	77.93%
Arc-Cosine [25]	***83.95%***	**94.50%**	***51.46%***	77.18%	26.80%	69.59%	***98.18%***	**99.09%**	43.06%	70.37%	38.50%	70.89%
Arc-Hybrid	**84.40%**	**94.50%**	**53.88%**	***77.91%***	30.67%	69.85%	**98.64%**	**99.09%**	***47.22%***	73.61%	38.50%	69.01%
CL-Cosine	77.06%	92.66%	46.36%	77.18%	***42.27%***	***71.13%***	95.00%	97.73%	46.76%	***78.24%***	***48.83%***	***80.28%***
CL-Hybrid	77.06%	94.04%	48.06%	**79.37%**	**42.53%**	**71.39%**	95.00%	97.73%	**47.69%**	**79.63%**	**50.23%**	**80.75%**

Table 4.1: Verification results for MBGC and FOCS datasets

algorithm.

- **Hybrid:** We compute the similarity scores by fusing the scores from SDL and LDDL.

- **CL-:** Prefix of using features from the ResNet-101 network trained by crystal loss [79].

4.4.2 Evaluation Results

MBGC: In the MBGC protocol, the verification task is specified by two sets: target and query. The protocol requires the algorithm to match each target sequence with all query sequences. Three verification experiments are defined: walking-vs-walking

(WW), activity-vs-activity (AA) and activity-vs-walking (AW). The verification results for the MBGC dataset are shown in Table 4.1 and Figure 4.2. We compare our method with the baseline algorithms and [20] using raw pixels as features ($\mathbf{DFRV}_{px}$) .The results of [20] are not included because they did not provide exact numbers in their paper. We also apply our method on features from the state-of-the-art network ArcFace [25]; results from ArcFace are shown with the prefix **Arc-**.

FOCS: Like MBGC, FOCS specifies three verification protocols: walking-vs-walking, activity-vs-walking, and activity-vs-activity. In these experiments, 481 walking videos and 477 activity videos are chosen as query videos. The size of target sets ranges from 109 to 135 video sequences. O'Toole et al. [70] evaluated the accuracy of humans recognizing people in the UT Dallas dataset. Human performance was reported for both through static and dynamic presentations of faces and bodies. The verification results of FOCS dataset are shown in Table 4.1 and Figure 4.3. Here we also compare our method with [20] in raw pixels as $\mathbf{DFRV}_{px}$. In the figures, *Human* refers to human performance with all bodies of target subjects seen and *Human_Face* refers to performance that only faces of the target subjects are seen. Similarly, the results of [20] and human performance are not included since they didn't provide exact numbers. Similar to MBGC, we also apply our method on the ArcFace features, the results of which are shown with the prefix **Arc-**.

IJB-A: Table 4.2 shows the verification results for IJB-A dataset. In this dataset, our method is compared with results in [1, 7, 65, 86, 93, 116].

From the results on the MBGC and FOCS datasets, we observe the following:

Methods	TAR@FAR		
	0.1%	1%	10%
[1]	-	78.70%	91.10%
[65]	72.50%	88.60%	-
[86]	81.30%	90.00%	96.40%
[7]	92.10%	96.80%	**99.00%**
[116]	92.00%	96.20%	***98.90%***
[93]	**95.25%**	**97.50%**	-
Cosine	76.95%	88.73%	96.04%
$DFRV_{deep}$	58.55%	83.31%	93.83%
RE	64.63%	85.37%	94.35%
AVDL	34.86%	81.44%	94.83%
SDL	78.00%	89.60%	96.32%
LDDL	78.58%	89.67%	96.51%
Hybrid	78.30%	89.65%	96.45%
CL-Cosine	94.73%	97.01%	98.46%
CL-Hybrid	***95.04%***	***97.18%***	98.56%

Table 4.2: Verification results for the IJB-A dataset

- In general, the proposed hybrid dictionary learning and matching approach performs better than cosine similarity, and reconstruction error-based methods, which shows the effectiveness of 1) learning discriminative information using structural and dynamical dictionaries which leverages the correlation of faces in videos and 2) matching video pairs using the subspace-to-subspace similarity metric, without any extra training data.

- LDDL performs better than AVDL consistently, which implies that the proposed method learns improved dynamical dictionaries than [111].

- In general, subspace-to-subspace similarity metric performs better than the reconstruction error-based metric, especially at low FARs, which shows that subspace-to-subspace similarity metric is more robust in difficult cases.

- The hybrid approach achieves better performance than single SDL and LDDL approaches in general. This implies that since the dictionaries are learned in different ways and capture different information, the error patterns of similarity scores computed from different dictionaries are complementary. Thus fusion can make consistent improvement on different datasets.

- Since [20] learns dictionaries from the raw pixels, the ROC curve is close to random guess in challenging protocols like activity-vs-activity and activity-vs-walking. In contrast, much better performance can be achieved by using the same algorithm using deep features, or even the cosine similarity between deep features, which shows the discriminative power of deep representations compared to raw pixels. In addition, the proposed approach can further improve the performance by exploiting the structural and temporal information to learn a more robust representation.

- The results using crystal loss features are comparable to the results using ArcFace features. Crystal loss features perform better than ArcFace features in more challenging protocols like activity-vs-walking and activity-vs-activity.

The proposed method shows consistent improvements on both features.

For the IJB-A dataset, the proposed method using crystal loss features performs better than other methods and baselines and is comparable to state-of-the-art result recently reported in [93]. The margin between the proposed method and the baseline methods in IJB-A is smaller compared to the MBGC and FOCS dataset because of the following reason: the videos in this protocol are much shorter than the MBGC and FOCS datasets since they consist of only I-frames. Also, some of the videos contain scene changes which cause the temporal correlation between frames to be much weaker than the MBGC and FOCS datasets. So it is difficult for the dynamical dictionary to extract helpful temporal information.

We wish to point out that, our dictionary learning algorithm usually converges within 20 iterations. The required number of atoms for each dictionary is small, which makes it computationally efficient. Since traditional reconstruction error-based methods like [20] perform inference of the sparse code (involves OMP) between every video pair, they are more time-consuming than the proposed approach which computes subspace-to-subspace similarity between videos.

4.5 Concluding Remarks

In this chapter, we proposed a dictionary learning and matching approach using deep representations for unconstrained video-based face verification. The proposed method learns structural and dynamical dictionaries from faces in video frames. A subspace-to-subspace similarity metric is defined for comparisons between

videos. We evaluated our approach on three video datasets. The experiment results demonstrate the effectiveness of the proposed approach.

Chapter 5: Uncertainty Modeling of Contextual-Connections between Tracklets for Unconstrained Video-based Face Recognition

5.1 Introduction

In this chapter, we continue to study the problem of video-based face recognition. Let us first look at a face recognition example in IJB-S. As shown in Figure 5.1, a single face is hard to recognize. But by utilizing the contextual information like body appearance, we may use the identity information obtained from the frontal face S_4 to help recognize the profile face S_1, which is very difficult to recognize otherwise. Thus an effective idea to improve the performance for unconstrained video-based face recognition is to leverage some video contextual information, such as body appearance and spatial-temporal correlation between person instances, to propagate the identity information from high-quality faces to low-quality ones.

This idea has been explored using graph-based approaches [30, 46, 92]. Graphs are constructed with nodes to represent one or more frames (tracklets) of person instances and edges to connect tracklets. However, a major limitation of these approaches is that their graphs are pre-defined and the edges are fixed during informa-

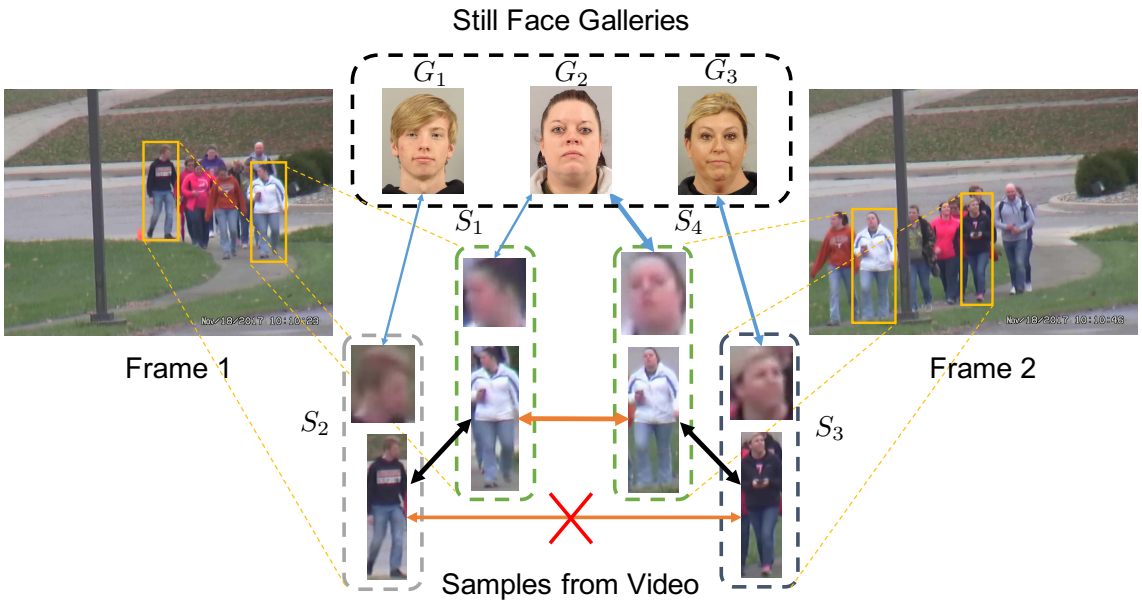


Figure 5.1: An example of video-based face recognition problem consisting of three still face gallery subjects and four samples from the videos. Orange arrows show positive connections from body appearance similarity. **Black arrows** indicate negative connections constructed from co-occurrence information. Blue arrows represent the facial similarities to the ground truth galleries. The thicker the arrows, the stronger the connections. The red cross indicates an misleading connection. A graph with fixed connections may propagate erroneous information through these misleading connections. (The figure is best viewed in color.)

tion propagation. A misleading connection may propagate erroneous information. As shown in Figure 5.1, these methods may propagate the identity information between S_2 and S_3 based on their similar body appearance, which might lead to erroneous propagation.

To address the problem, we propose a graphical-model-based framework called Uncertainty-Gated Graph (UGG) to model the uncertainty of connections built us-

ing contextual information. We formulate UGG as a conditional random field on the graph with additional gate nodes introduced on the connected graph edges. With a carefully designed energy function, the identity distribution of tracklets[1] is updated by the information propagated through these gate nodes during inference. In turn, these gate nodes are adaptively updated according to the identity distributions of the connected tracklets. The uncertainty gate nodes consist of two types of gates: positive gates that control the confidence of the positive connections (encourage the connected pairs to have the same identity) and negative gates that control negative ones (discourage pairs to have the same identity). It is worth noting that negative connections can significantly contribute to performance improvements by discouraging similar identity distribution between clearly distinct subjects, e.g., two people in the same frame[2]. Explicitly modeling positive/negative information separately allows our model to consider different contextual information in challenging conditions, and leads to improved uncertainty modeling.

Our approach can be directly applied at inference time, or plugged onto an end-to-end network architecture for supervised and semi-supervised training. The proposed method is evaluated on two challenging datasets, the Cast Search in Movies (CSM) dataset [46] and the IARPA Janus Surveillance Video Benchmark (IJB-S) dataset [52] and shown to yield superior performance compared to existing methods.

[1]We follow the same definition of tracklets with [46].

[2]In Figure 5.1, the co-occurrence of S_3 and S_4 in the same frame of the video is a strong prior to indicate their different identities.

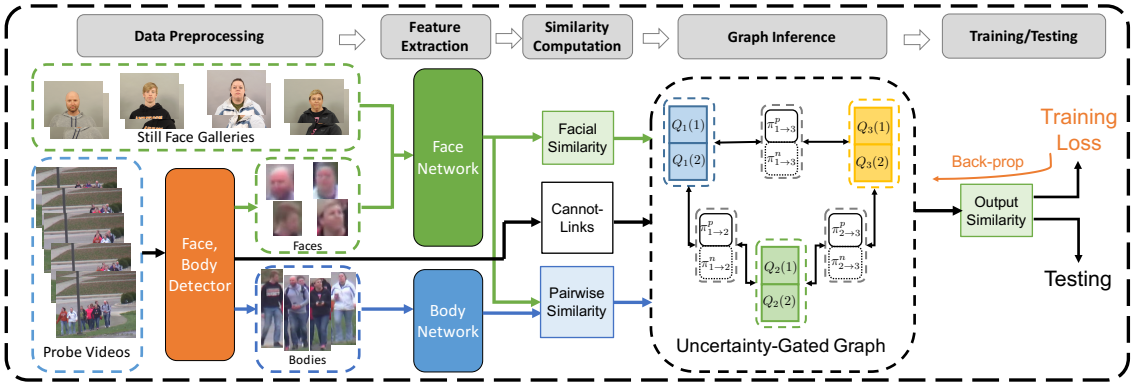


Figure 5.2: Overview of the proposed method. Given still face galleries and probe videos, we first detect all the faces and corresponding bodies from the videos. Faces are associated into tracklets by a tracker. Face features for galleries and tracklets, and body features for tracklets are extracted by corresponding networks. Similarities are computed from these flattened features. Facial and body similarities, together with cannot-link constrains from the detection information are fed into the proposed UGG model. After inference, the output is used for testing, or generating the loss for end-to-end training.

5.2 Related Work

Label Propagation: Label propagation [132] has many applications in computer vision. Huang *et al.* [46] proposed an approach for searching person in videos using a label propagation scheme instead of trivial label diffusion. Kumar *et al.* [58] proposed a video-based face recognition method by selecting key-frames and propagating the labels on key-frames to other frames. Sheikh *et al.* [91] used label propagation to reduce the runtime for semantic segmentation using random forests. Tripathi *et al.* [103] introduced a label propagation-based object detection method.

Conditional Random Field: The Conditional Random Field (CRF) [59]

is a commonly used probabilistic graphical models in computer vision research. Krähenbühl *et al.* [56] is one of the early researchers to use CRF for semantic segmentation. Chen *et al.* [18, 19] proposed a DCNN-based system for semantic segmentation and used a CRF for post-processing. Zheng *et al.* [129] further introduced an end-to-end framework of a deep network with a CRF module for semantic segmentation. Du *et al.* [30] used a CRF to solve the face association problem in unconstrained videos.

Graph Neural Networks: A Graph Neural Network (GNN) [42, 88] is a neural network combined with graphical models such that messages are passed in the graph to update the hidden states of the network. Shen *et al.* [92] used a GNN for person re-identification problem. Hu *et al.* [44] introduced a structured label prediction method based on a GNN, which allows positive and negative messages to pass between labels guided by external knowledge. But the graph edges are fixed during testing. Wang *et al.* [110] introduced a zero-shot learning method using stacked GNN modules. Lee *et al.* [60] proposed another multi-label zero-shot learning method by message passing in a GNN based on knowledge graphs.

Most of the graph-based methods mentioned above only allow positive messages to pass in the graph, and all of them rely on graphs with fixed edges during testing.

5.3 Method

The overview of the method is shown in Figure 5.2. For each probe video, faces are detected and associated into tracklets. Initial facial similarities between gallery images and probe tracklets are computed by a still face recognizer. Connections between tracklets are generated based on the similarity of their facial, body appearances and their spatio-temporal relationships. Then, we build the UGG where these tracklets and connections act as nodes and edges. The connections between tracklets are modeled as uncertainty gates between nodes. The inference can be efficiently implemented by message passing to optimize the energy function of the UGG module.

5.3.1 Problem Formulation

For a video-based face recognition problem, suppose we have C gallery subjects and a probe video. The faces in this video are first detected and tracked into N tracklets. For each tracklet, we compute C similarity scores to gallery subjects.

Suppose we are given the gallery-to-tracklet similarity $\mathbf{S}^{gt} = \left[s_{li}^{gt} \right] \in \mathbb{R}^{C \times N}$ and the tracklet-to-tracklet similarity $\mathbf{S}^{tt} = \left[s_{ij}^{tt} \right] \in \mathbb{R}^{N \times N}$, where s_{li}^{gt} is the similarity between the gallery l and the tracklet i, s_{ij}^{tt} is the similarity between tracklet i and j. Furthermore, a cannot-link matrix $\mathbf{L}^{tt} = \left[L_{ij}^{tt} \right] \in \{0,1\}^{N \times N}$ is given such that

$$L_{ij}^{tt} = \begin{cases} 1 & \text{identities of tracklet } i \text{ and } j \text{ are different} \\ 0 & \text{no constraint} \end{cases} \tag{5.1}$$

A cannot-link exists between tracklet i and j if they absolutely do not belong to same gallery subject.

Here, $\mathbf{S}^{gt}$ provides prior identity information, $\mathbf{S}^{tt}$ provides the positive contextual information between tracklets and $\mathbf{L}^{tt}$ provides the negative contextual information. By combining these information, the output gallery-to-tracklet similarity is computed as

$$\tilde{\mathbf{S}}^{gt} = UGG(\mathbf{S}^{gt}, \mathbf{S}^{tt}, \mathbf{L}^{tt}) \in \mathbb{R}^{C \times N} \tag{5.2}$$

where $UGG(\cdot)$ is a function based on the proposed Uncertainty-Gated Graph. In the following sections, we introduce the model in detail.

5.3.2 Uncertainty-Gated Graph

First, given a video with N tracklets detected, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is built where each node corresponds to a tracklet. Node i is only connected to its neighbors $\mathcal{N}(i)$. Based on the graph $\mathcal{G}$, we define a random field $\mathbf{X} = \{X_1, \ldots, X_N\}$ associated to nodes $\mathcal{V}$. $X_i \in \mathcal{L} = \{1, \ldots, C\}$ is the label variable of tracklet i. $X_i = l$ means gallery subject l is assigned to tracklet i. We call these nodes as *sample nodes*.

We further add *gates nodes* to each of the edges in $\mathcal{E}$ attached with a random field $\mathbf{Y} = \{Y^p_{i \to j}, Y^n_{i \to j}\}$. In each gate node $i \to j$, we place two gate variables, the *positive gate* $Y^p_{i \to j} \in \{0, 1\}$ and the *negative gate* $Y^n_{i \to j} \in \{0, 1\}$, to control the connections between tracklets i and j.

5.3.2.1 Energy Function

The energy function of the UGG module is defined as

$$E(\mathbf{x}, \mathbf{y}) = \sum_{i \in \mathcal{V}} \psi_u^x(x_i) + \sum_{i \in \mathcal{V}, j \in \mathcal{N}(i)} [\psi_u^p(y_{i \to j}^p) + \psi_u^n(y_{i \to j}^n) + \psi_t^p(x_i, x_j, y_{i \to j}^p) + \psi_t^n(x_i, x_j, y_{i \to j}^n)] \tag{5.3}$$

The unary potential for tracklet i is defined based on the identity information $\mathbf{S}^{gt}$ as

$$\psi_u^x(x_i = l) = -T_{gt} \cdot s_{li}^{gt} \tag{5.4}$$

where T_{gt} is the temperature factor. The penalty will be low if identity information s_{li}^{gt} is strong.

We also define the unary potential for the positive gate based on relationship information $\mathbf{S}^{tt}$ as

$$\psi_u^p(y_{i \to j}^p = 1) = -T_{tt} \cdot s_{ij}^{tt} \tag{5.5}$$

where T_{tt} is the corresponding temperature factor. Penalty of an open positive gate at edge $i \to j$ will be low if positive connection s_{ij}^{tt} is strong.

The unary potential for the negative gate is defined as

$$\psi_u^n(y_{i \to j}^n = k) = \begin{cases} 0 & \text{if } L_{ij}^{tt} = k \\ +\infty & \text{otherwise} \end{cases} \tag{5.6}$$

for $k \in \{0, 1\}$. Therefore, opening of the negative gate at node $i \to j$ is determined by the negative connection L_{ij}^{tt}.

The positive triplet potential is defined as

$$\psi_t^p(x_i, x_j, y_{i\to j}^p) = \begin{cases} \alpha_p & \text{if } y_{i\to j}^p = 1 \text{ and } x_i \neq x_j \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

where α_p is the positive penalty. Since $y_{i\to j}^p = 1$ means an open positive gate between tracklet i and j, it generates positive information to nodes i and j if x_i and x_j take different labels.

Similarly, the negative triplet potential is defined as

$$\psi_t^n(x_i, x_j, y_{i\to j}^n) = \begin{cases} \alpha_n & \text{if } y_{i\to j}^n = 1 \text{ and } x_i = x_j \\ 0 & \text{otherwise} \end{cases} \tag{5.8}$$

where α_n is the negative penalty. Since $y_{i\to j}^n = 1$ means an open negative gate between tracklet i and j, it generate negative information to nodes i and j if x_i and x_j have the same label.

5.3.3 Model Inference

Directly looking for the label assignment that minimizes $E(\mathbf{x}, \mathbf{y})$ is a combinatorial optimization problem which is intractable. Instead, similar to [56], we use the mean field method to approximate the distribution $P(\mathbf{X}, \mathbf{Y}) \propto \exp(-E(\mathbf{X}, \mathbf{Y}))$ by the product of independent marginals

$$Q(\mathbf{X}, \mathbf{Y}) = \prod_i Q_i(X_i) \prod_{j\in\mathcal{N}(i)} Q_{i\to j}^p(Y_{i\to j}^p) Q_{i\to j}^n(Y_{i\to j}^n) \tag{5.9}$$

Here $Q_i(X_i)$ is the identity distribution of node i, $Q_{i\to j}^p(Y_{i\to j}^p)$ and $Q_{i\to j}^n(Y_{i\to j}^n)$ are the status distributions of positive and negative gates on edge $i \to j$ respectively.

Minimizing the KL-divergence $D(Q||P)$ between $P(\mathbf{X},\mathbf{Y})$ and $Q(\mathbf{X},\mathbf{Y})$ yields the following updating equations:

1) For the *tracklet nodes*, we have

$$\begin{aligned}
&Q_i^{(t)}(x_i=l)\\
=&\frac{1}{Z_i}\exp\left\{-\psi_u^x(l)-\sum_{j\in\mathcal{N}(i)}\sum_{l'}\sum_{k\in\{0,1\}}\psi_t^p(l,l',k)Q_j^{(t-1)}(l')Q_{i\to j}^{p,(t-1)}(k)-\right.\\
&\left.\sum_{j\in\mathcal{N}(i)}\sum_{l'}\sum_{k\in\{0,1\}}\psi_t^n(l,l',k)Q_j^{(t-1)}(l')Q_{i\to j}^{n,(t-1)}(k)\right\}\\
=&\frac{1}{Z_i}\exp\left\{-\psi_u^x(l)-\alpha_p\sum_{j\in\mathcal{N}(i)}Q_{i\to j}^{p,(t-1)}(1)\sum_{l'\neq l}Q_j^{(t-1)}(l')-\alpha_n\sum_{j\in\mathcal{N}(i)}Q_{i\to j}^{n,(t-1)}(1)Q_j^{(t-1)}(l)\right\}\\
=&\frac{1}{Z_i}\exp\left\{T_{gt}s_{li}^{gt}+\alpha_p\sum_{j\in\mathcal{N}(i)}Q_{i\to j}^{p,(t-1)}(1)Q_j^{(t-1)}(l)-\alpha_n\sum_{j\in\mathcal{N}(i)}Q_{i\to j}^{n,(t-1)}(1)Q_j^{(t-1)}(l)\right\}
\end{aligned}\tag{5.10}$$

where Z_i is the normalization factor and $Q^{(t)}(\cdot)$ is the approximated distribution at the t-th iteration. It is initialized by

$$Q_i^{(0)}(x_i=l)=\frac{1}{Z_i}\exp\{T_{gt}s_{li}^{gt}\}\tag{5.11}$$

2) For the *positive gates*, we have

$$\begin{aligned}
Q_{i\to j}^{p,(t)}(y_{i\to j}^p=1)=&\frac{1}{Z_{i\to j}^p}\exp\left\{-\psi_u^p(1)-\sum_{l,l'}\sum_{j\in\mathcal{N}(i)}\psi_t^p(l,l',1)Q_i^{(t-1)}(l)Q_j^{(t-1)}(l')\right\}\\
=&\frac{1}{Z_{i\to j}^p}\exp\left\{-\psi_u^p(1)-\alpha_p\sum_{l'\neq l}Q_i^{(t-1)}(l)Q_k^{(t-1)}(l')\right\}\\
=&\frac{1}{Z_{i\to j}^p}\exp\left\{T_{tt}s_{ij}^{tt}+\alpha_p\sum_{l}Q_i^{(t-1)}(l)Q_j^{(t-1)}(l)-\alpha_p\right\}
\end{aligned}\tag{5.12}$$

For normalization purpose, we set the factor $Z_{i\to j}^p$ so that $\sum_{j\in\mathcal{N}(i)}Q_{i\to j}^{p,(t)}(1)=1$.

Thus we have

$$Q_{i\to j}^{p,(t)}(y_{i\to j}^{p}=1)=\frac{1}{Z_{i\to j}^{p}}\exp\left\{T_{tt}s_{ij}^{tt}+\alpha_p\sum_{l}Q_i^{(t-1)}(l)Q_j^{(t-1)}(l)\right\} \tag{5.13}$$

It is initialized by

$$Q_{i\to j}^{p,(0)}(y_{i\to j}^{p}=1)=\frac{1}{Z_{i\to j}^{p}}\exp\{T_{tt}s_{ij}^{tt}\} \tag{5.14}$$

3) For the *negative gates*, we have

$$\begin{aligned}Q_{i\to j}^{n,(t)}(y_{i\to j}^{n}=1)=&\frac{1}{Z_{i\to j}^{n}}\exp\left\{-\psi_u^n(1)-\sum_{l,l'}\sum_{j\in\mathcal{N}(i)}\psi_t^n(l,l',1)Q_i^{(t-1)}(l)Q_j^{(t-1)}(l')\right\}\\ =&\frac{1}{Z_{i\to j}^{n}}\exp\left\{-\psi_u^n(1)-\alpha_n\sum_{l}Q_i^{(t-1)}(l)Q_j^{(t-1)}(l)\right\}\end{aligned} \tag{5.15}$$

Since

$$\psi_u^n(y_{i\to j}^{n}=k)=\begin{cases}0 & \text{if } L_{ij}^{tt}=k\\ +\infty & \text{otherwise}\end{cases} \tag{5.16}$$

for $k\in\{0,1\}$, we have

$$Q_{i\to j}^{n,(t)}(y_{i\to j}^{n}=k)=\begin{cases}k & \text{if } L_{ij}^{tt}=1\\ 1-k & \text{otherwise}\end{cases} \tag{5.17}$$

for $k\in\{0,1\}$, $t=0,\ldots,K$.

Let $\mathbf{q}_i^{(t)}=\begin{bmatrix}Q_i(1)^{(t)} & \ldots & Q_i(C)^{(t)}\end{bmatrix}^T$ be the identity distribution vector of node i at the t-th iteration. $\pi_{i\to j}^{p,(t)}=Q_{i\to j}^{p,(t)}(1)$ and $\pi_{i\to j}^{n,(t)}=Q_{i\to j}^{n,(t)}(1)$ be the probability of opened positive and negative gates on edge $i\to j$ respectively, we have the following message passing equations:

1) For *sample nodes*, we have

$$\begin{aligned}\mathbf{q}_i^{(0)} &= \text{softmax}(T_{gt}\mathbf{S}_{:,i}^{gt}) \\ \mathbf{q}_i^{(t)} &= \text{softmax}(T_{gt}\mathbf{S}_{:,i}^{gt} + \alpha_p \sum_{j\in\mathcal{N}(i)} \pi_{i\to j}^{p,(t-1)}\mathbf{q}_j^{(t-1)} - \alpha_n \sum_{j\in\mathcal{N}(i)} \pi_{i\to j}^{n,(t-1)}\mathbf{q}_j^{(t-1)})\end{aligned} \tag{5.18}$$

where $\mathbf{S}_{:,i}^{gt}$ is the ith column of $\mathbf{S}^{gt}$.

2) For *gate nodes*, we let the marginal distribution of positive gates $\sum_{j\in\mathcal{N}(i)} \pi_{i\to j}^{p,(t)} = 1$ for normalization purpose. Then we have

$$\begin{aligned}\pi_{i\to j}^{p,(0)} &= \underset{\mathcal{N}(i)}{\text{softmax}}(T_{tt}s_{ij}^{tt}) \\ \pi_{i\to j}^{p,(t)} &= \underset{\mathcal{N}(i)}{\text{softmax}}(T_{tt}s_{ij}^{tt} + \alpha_p \mathbf{q}_i^{(t-1)} \cdot \mathbf{q}_j^{(t-1)})\end{aligned} \tag{5.19}$$

where $\text{softmax}_{\mathcal{N}(i)}(\cdot)$ is the softmax operation in the neighborhood $\mathcal{N}(i)$. From (5.6), we also have

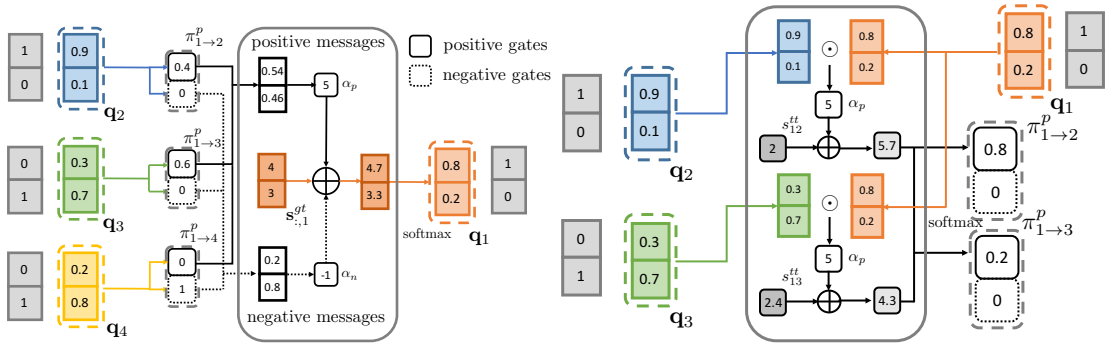
$$\pi_{i\to j}^{n,(t)} = L_{ij}^{tt} \tag{5.20}$$

for $t = 0, \ldots, K$. Thus, the marginal probability of a negative gate is fixed during inference.

Two illustrations of message passing and node update are shown in Figure 5.3. From these recursive updating equations we can see that:

1) When updating *sample node* i, identity information from $\mathbf{q}_j$ in $\mathcal{N}(i)$ is propagated through positive gate $\pi_{i\to j}^{p}$ and negative gate $\pi_{i\to j}^{n}$ and collected as positive (α_p) and negative $(-\alpha_n)$ message, respectively. These messages together with the prior identity information $\mathbf{S}_{:,i}^{gt}$ are combined to update $\mathbf{q}_i$, the identity distribution of node i, in the next iteration.

2) When updating *gate node* $i \to j$, the identity similarity between $\mathbf{q}_i$ and



(a) Update Sample Distributions (b) Update Gate Probabilities

Figure 5.3: (a) shows the update of $\mathbf{q}_1$. Distribution of the neighbors are weighted by the probability of opening gates and collected as positive and negative messages, respectively. The new marginal distribution is updated by the sum of messages and the unary scores. Grey boxes are the ground truth labels of samples. (b) shows the update of gate $\pi^p_{1\to 2}$ and $\pi^p_{1\to 3}$. Distributions of sample node pairs are used to modify the marginal probability of positive gates. We can see that the connection between sample 1 and 3 is misleading since s^{tt}_{13} is large but they belong to different identities. After updating the probability of gates by utilizing the information from neighboring nodes, $\pi^p_{1\to 3}$ drops comparing to (a), results in less positive information passing between sample 1 and 3 in the next iteration. $\odot$ is inner product operation.

its neighbor $\mathbf{q}_j$ in $\mathcal{N}(i)$ is measured by pairwise inner product. By combining this similarity with the initial contextual connection score s^{tt}_{ij}, the probability of gate openness $\pi^p_{i\to j}$ for the positive gate is updated. If $\mathbf{q}_i \cdot \mathbf{q}_j$ is small, $\pi^p_{i\to j}$ will gradually vanish in iterations, which avoids misleading connections propagating erroneous information. Negative gates based on cannot-links are fixed during inference.

We conduct these bidirectional updates jointly so that the samples nodes receive useful information from their neighbors through reliable connections to gradu-

ally refine their identity distributions, and the misleading connections in the graph are gradually corrected by these refined identity distributions in return.

After obtaining the approximation $Q(\mathbf{X}, \mathbf{Y})$ that minimizes $D(Q||P)$ in K iterations, we use the identity distribution $\mathbf{q}_i^{(K)}$ as the output similarity scores $\tilde{\mathbf{S}}_{:,i}^{gt}$ from tracklet i to gallery subjects.

5.3.4 UGG: Training and Testing Settings

Testing with UGG: For testing, the UGG module can be directly applied at inference time, where we compute input matrices $\mathbf{S}^{gt}$, $\mathbf{S}^{tt}$ and $\mathbf{L}^{tt}$ from the video, setting the hyperparameters in the UGG module. Then the module produces the output similarity $\tilde{\mathbf{S}}^{gt}$ by recursive forward calculations.

Training with UGG: Similar to RNN, the proposed UGG module can be considered as a differentiable recurrent module and be inserted into any neural networks for end-to-end training. If video face training data is available, we can utilize them for training to further improve the performance.

Given tracklets $\{T_i\}$ from a training video and galleries $\{G_l\}$, we use two DCNN networks F_{gt} and F_{tt} with parameters $\boldsymbol{\theta}_{gt}$ and $\boldsymbol{\theta}_{tt}$ pretrained on still images to generate $\mathbf{S}^{gt}$ and $\mathbf{S}^{tt}$ respectively as

$$s_{li}^{gt} = F_{gt}(G_l, T_i; \boldsymbol{\theta}_{gt}), \quad s_{ij}^{tt} = F_{tt}(T_i, T_j; \boldsymbol{\theta}_{tt}) \tag{5.21}$$

and feed into the UGG module.

After the module generates output similarity $\tilde{\mathbf{S}}^{gt} = \left[\tilde{\mathbf{s}}_1, \ldots, \tilde{\mathbf{s}}_N\right]$ after K iter-

ations, we compute the loss of this video as

$$L = \frac{1}{N}\sum_{i\in\mathcal{S}} L_C(\tilde{\mathbf{s}}_i, z_i^c) + \lambda\frac{1}{N^2}\sum_{i,j\in\mathcal{S}} L_P(s_{ij}^{tt}, z_{ij}^b) \tag{5.22}$$

Here, L_C is a cross-entropy loss on $\tilde{\mathbf{s}}_i$ with ground truth classification label z_i^c. L_P is a pairwise binary cross-entropy loss on s_{ij}^{tt} with ground truth binary label z_{ij}^b. λ is the weight factor. $\mathcal{S}$ is the set of labeled tracklets.

Back-propagation through the whole networks on the overall loss L is used to learn the DCNN parameters $\boldsymbol{\theta}_{gt}$, $\boldsymbol{\theta}_{tt}$ in F_{gt} and F_{tt}, together with the temperature parameters T_{gt}, T_{tt} in the UGG module. T_{gt}, T_{tt} are learned in order to find a good balance between the unary scores and the messages from the neighbors during updates.

Depending on the different choices of $\mathcal{S}$, the training can be categorized into three settings:

1. Supervised Setting: $\mathcal{S} = \mathcal{V}$, where every training sample in the graph is labeled. In this setting, we can directly utilize all the tracklets in the graph for training.

2. Semi-Supervised Setting: $\varnothing \subset \mathcal{S} \subset \mathcal{V}$, where training samples in the graph are only partially labeled. In this setting, the output of the module still depends on all the tracklets in the graph through information propagation. Thus, via back-propagation, the supervision information is propagated from labeled tracklets to unlabeled tracklets through the connections in the UGG module and enable them to benefit the training.

3. Unsupervised Setting: $\mathcal{S} = \varnothing$, where no labeled training data is avail-

able. In this setting, we skip the training part since no supervision is provided.

5.4 Experiments

In this section, we report experiment results of the proposed method in two challenging video-based person search and face recognition datasets: the Cast Search in Movies (CSM) dataset and the IARPA Janus Surveillance Video Benchmark (IJB-S) dataset.

5.4.1 Datasets

CSM: The CSM dataset is a large-scale person search dataset comprising a query set containing cast portraits in still images and a gallery set containing tracklets collected from movies. The evaluation metrics of the dataset include *mean Average Precision* (mAP) and recall of the tracklet identification (R@k). Two protocols are used in the CSM dataset. One is IN which only search among tracklets in a single movie once a time. Another is ACROSS which search among tracklets in all the movies in the testing set. Please refer [46] for more details.

IJB-S: In this chapter, we mainly focus on two protocols related to our topic, the surveillance-to-single protocol (S2SG) and the surveillance-to-booking protocol (S2B). We report the per tracklet average top-K identification accuracy and the End-to-End Retrieval Rate (EERR) metric proposed in [52] for performance evaluation.

Methods	IN				ACROSS			
	mAP	R@1	R@3	R@5	mAP	R@1	R@3	R@5
FACE(avg)	53.33%	76.19%	91.11%	96.34%	42.16%	53.15%	61.12%	64.33%
PPCC(avg) [46]	62.37%	84.31%	94.89%	98.03%	59.58%	63.26%	74.89%	78.88%
PPCC(max) [46]	63.49%	83.44%	94.40%	97.92%	62.27%	62.54%	73.86%	77.44%
UGG-U(avg)	62.81%	85.21%	95.65%	98.30%	63.31%	66.73%	76.09%	79.32%
UGG-U(max)	63.74%	84.93%	95.36%	**98.37%**	63.42%	65.72%	74.90%	77.88%
UGG-U(favg)	64.36%	84.96%	94.90%	97.98%	64.85%	67.33%	75.38%	78.21%
UGG-ST(favg)	65.12%	86.73%	95.70%	98.34%	67.00%	71.16%	77.82%	80.15%
UGG-T(favg)	**65.41%**	**87.28%**	**95.87%**	98.28%	**67.60%**	**71.51%**	**78.33%**	**80.56%**

Table 5.1: Results on CSM dataset. Notice that *UGG-U(favg)* is the unsupervised, initial setting before training. *UGG-ST(favg)* is the semi-supervised training setting with 25% samples labeled. *UGG-T(favg)* is the supervised training setting.

5.4.2 Implementation Details

5.4.2.1 CSM: Pre-processing details

For the CSM dataset, we use the 256-dimensional facial and body features provided by [46]. We first flatten both facial and body features in each tracklet by average pooling. Denote facial features for galleries as $\mathbf{F}_F^g$, flattened facial features for tracklets as $\mathbf{F}_F^t$ and flattened body features for tracklets as $\mathbf{F}_B^t$. Three linear embedding matrices $\mathbf{W}_F^{gt}$, $\mathbf{W}_F^{tt}$, $\mathbf{W}_B^{tt}$, all with size 256×256 are applied on the features respectively for more discriminative representation.

We use the cosine similarity between $\mathbf{W}_F^{gt}\mathbf{F}_F^g$ and $\mathbf{W}_F^{gt}\mathbf{F}_F^t$ as the gallery-to-tracklet similarity $\mathbf{S}^{gt} = \mathbf{S}_{F,cos}^{gt}$. To improve the reliability of positive connections,

Methods	Top-K Average Accuracy *with Filtering*						EERR metric *without Filtering*					
	R@1	R@2	R@5	R@10	R@20	R@50	R@1	R@2	R@5	R@10	R@20	R@50
FACE(favg)	64.86%	70.87%	77.09%	81.53%	86.11%	93.24%	29.62%	32.34%	35.60%	38.36%	41.53%	46.78%
PPCC(favg) [46]	67.31%	73.21%	79.06%	83.12%	87.38%	93.68%	30.57%	33.28%	36.53%	39.10%	42.00%	47.00%
FACE(sub) [126]	69.82%	75.38%	80.54%	84.36%	87.91%	94.34%	32.43%	34.89%	37.74%	40.01%	42.77%	47.60%
UGG-U(favg)	74.20%	77.67%	81.43%	84.54%	87.96%	93.62%	32.70%	35.04%	37.54%	39.79%	42.43%	47.10%
UGG-U(sub)	**77.59%**	**80.46%**	**83.70%**	**86.20%**	**89.23%**	**94.55%**	**34.79%**	**36.88%**	**39.11%**	**40.90%**	**43.37%**	**47.86%**

Table 5.2: 1:N Search results of IJB-S surveillance-to-single protocol. *UGG-U(favg)* directly uses the cosine similarities between average-flattened features. *UGG-U(sub)* uses the subspace-subspace similarity proposed in [126].

we use the fusion of the cosine similarities between $\mathbf{W}_F^{tt}\mathbf{F}_F^t$ and between $\mathbf{W}_B^{tt}\mathbf{F}_B^t$ as the tracklet-to-tracklet similarity $\mathbf{S}^{tt} = \lambda_f \mathbf{S}_{F,cos}^{tt} + (1-\lambda_f)\mathbf{S}_{B,cos}^{tt}$, with fusion weight λ_f. No detection information is provided in this dataset so the cannot-link matrix $\mathbf{L}^{tt}$ is all-zero. We feed $\mathbf{S}^{gt}$, $\mathbf{S}^{tt}$ and $\mathbf{L}^{tt}$ into the proposed UGG module. The module iterates for K iterations and produce the output similarity $\tilde{\mathbf{S}}^{gt}$.

5.4.2.2 CSM: Testing details

For testing, we use all the tracklets in each movie to build the graph. The neighborhood $\mathcal{N}(i)$ for tracklet i is defined as the top 10% of the tracklets in the movie with the largest tracklet-to-tracklet similarity score to tracklet i. We apply identity embedding matrices on the features and compute similarities. Then the UGG module is used to produce the output similarity scores $\tilde{\mathbf{S}}^{gt}$. Using the validation set, we choose parameters $T_{gt} = 10$, $T_{tt} = 15$, $\alpha_p = 5$, $K = 2$, $\lambda = 0.1$ and $\lambda_f = 0.1$ for the IN protocol and $T_{gt} = 20$, $T_{tt} = 30$, $\alpha_p = 15$, $K = 2$, $\lambda = 0.1$ and

Methods	Top-K Average Accuracy *with Filtering*						EERR metric *without Filtering*					
	R@1	R@2	R@5	R@10	R@20	R@50	R@1	R@2	R@5	R@10	R@20	R@50
FACE(favg)	66.48%	71.98%	77.80%	82.25%	86.56%	93.41%	30.38%	32.91%	36.15%	38.77%	41.86%	46.79%
PPCC(favg) [46]	68.96%	74.44%	79.84%	83.75%	87.68%	93.80%	31.37%	33.98%	37.04%	39.49%	42.35%	47.01%
FACE(sub) [126]	69.86%	75.07%	80.36%	84.32%	88.07%	94.33%	32.44%	34.93%	37.80%	40.14%	42.72%	47.58%
UGG-U(favg)	74.79%	78.35%	81.81%	84.85%	88.15%	93.80%	33.29%	35.48%	37.87%	40.02%	42.60%	47.14%
UGG-U(sub)	**77.02%**	**80.08%**	**83.39%**	**86.20%**	**89.29%**	**94.62%**	**34.83%**	**36.81%**	**39.11%**	**41.10%**	**43.38%**	**47.74%**

Table 5.3: 1:N Search results of IJB-S surveillance-to-booking protocol. *UGG-U(favg)* directly uses the cosine similarities between average-flattened features. *UGG-U(sub)* uses the subspace-subspace similarity proposed in [126].

$\lambda_f = 0.1$ for the ACROSS protocol.

5.4.2.3 CSM: Training details

For end-to-end training, we train the embedding matrices $\mathbf{W}_F^{gt}$, $\mathbf{W}_F^{tt}$, and $\mathbf{W}_B^{tt}$, together with temperatures T_{gt} and T_{tt} in the UGG module, implemented in PyTorch [73]. For each movie, we use all the galleries and randomly pick 1/8 of the tracklets to construct the graph. The overall loss is computed by (5.22). The network is trained using Adam solver [54] for 20 epochs with batch size 2 (2 movies in each batch). The initial learning rate is 1×10^{-4}. All embedding matrices are initialized as identity matrix. We initialize T_{gt} and T_{tt} by 10 and 15 respectively and fix other parameters as $\alpha_p = 5$, $K = 2$, $\lambda = 0.1$ and $\lambda_f = 0.1$ during training.

5.4.2.4 IJB-S: Pre-processing details

For the IJB-S dataset, we follow the pre-processing steps in [126]. We employ the multi-scale face detector DPSSD [79] to detect faces in surveillance videos. We use the facial landmark branch of All-in-One Face [82] as the fiducial detector. Face alignment is performed using the seven-point similarity transform. Similar to [126], we use a ResNet-101 [40] and a Inception-ResNet-v2 [99], both trained on the union of the MSCeleb-1M dataset [38], the UMDFaces dataset [4], and the UMDFaces Video dataset with the crystal loss [78], to represent the faces. A triplet probabilistic embedding (TPE) [86] trained on the UMDFaces dataset is applied on face features for dimensionality reduction to 128.

We also use the Mask R-CNN [39] implemented on Detectron [34] to detect the bodies in the videos and match each body to the face with the highest overlap ratio. The detected bodies are represented by a re-id network with ResNet-50 architecture trained on the Market1501 dataset [128], implemented on [130]. The network produces 2048-dimensional feature for each body.

We use SORT [5] to construct tracklets for every face appearing in the videos. Facial and body features are first flattened by average pooling for each gallery and tracklet. $\mathbf{S}^{gt}$ and $\mathbf{S}^{tt}$ are computed in the same way as the CSM dataset, except there is no embedding matrices applied since no training set available on IJB-S. We use the bounding box information from the detector to build the co-occurence cannot-link matrix $\mathbf{L}^{tt}$ such that all the tracklets with distinct bounding boxes appear in the same frame will have cannot-links between them.

5.4.2.5 IJB-S: Testing details

For the IJB-S dataset we empirically use the hyperparameter configuration of $T_{gt} = 15$, $T_{tt} = 15$, $\alpha_p = 10$, $\alpha_n = 2$, $K = 4$, $\lambda = 0.1$ and $\lambda_f = 0.1$ in the UGG module for testing. All the other details are the same as the CSM dataset.

To compare with [126], we use the same configurations for tracklets filtering and evaluation metrics for each configuration: 1) *with Filtering:* We keep those tracklets with length greater than or equal to 25 and average detection score greater than or equal to 0.9. 2) *without Filtering.*

5.4.3 Baseline Methods

We conduct experiments on the CSM and IJB-S dataset with two baseline methods: *FACE:* facial similarity is directly used without any refinement. *PPCC:* The Progressive Propagation via Competitive Consensus method proposed in [46] is used for post-processing. For the CSM dataset, we use the numbers reported in [46]. For the IJB-S dataset, we implement the method using the code provided by the author.

For fair comparisons, following [46], two settings of input similarity are used: *avg:* similarity is computed by the average of all frame-wise cosine similarities between a gallery and a tracklet, or two tracklets. *max:* similarity is computed by the maximum of all frame-wise cosine similarities between a gallery and a tracklet, or two tracklets. On IJB-S, we also implement the subspace-based similarity following [126], denoted as *sub*.

As introduced in Chapter 3, two recent works [35,36] have also reported results on the IJB-S dataset. These works built video templates by matching their detections with ground truth bounding boxes provided by the dataset. Our method follows [126] and associates faces across the video frames to build templates(tracklets) without utilizing any ground truth information. Since these two template building procedures are very different, a direct comparison is not meaningful.

Results of these baselines on two datasets are shown in Tables 5.1, 5.2 and 5.3 respectively. Average run time of *PPCC* is also reported in Table 5.4, on a machine with 72 Intel Xeon E5-2697 CPUs, 512GB of memory and two NVIDIA K40 GPUs. We observe that *PPCC* only achieves marginal improvements on the IJB-S dataset. Its speed is also slow during inference, especially when large graphs are constructed.

5.4.4 Evaluation on the Proposed UGG method

On the CSM dataset, depending on the usage of training data, we evaluate three settings of UGG including: *UGG-U:* without training, the UGG module works in *unsupervised setting* as post-processing module. *UGG-T:* with fully-labeled training data, the UGG module and linear embeddings are trained in *supervised setting*. *UGG-ST:* with 25% labeled and 75% unlabeled training data by random selection in each movie, the UGG module and linear embeddings and are trained in *semi-supervised setting*. On the IJB-S dataset, since the dataset only provide test data, we use the *unsupervised setting* and only test *UGG-U*.

The additional input similarity used for training is the cosine similarity be-

tween flattened features after average pooling and denoted as *favg*. Corresponding results are shown in Tables 5.1, 5.2 and 5.3 respectively, with average run time tested on the same machine reported in Table 5.4.

5.4.4.1 Observations on CSM

1. UGG vs FACE: All the settings of UGG perform significantly better than the raw baseline *FACE*. *UGG-T(favg)* provides state-of-the-art results on almost all the evaluation metrics with large margins, which demonstrates the effectiveness of the proposed method utilizing contextual connections.

2. UGG vs PPCC [46]: Using the same input similarity without training, *UGG-U* performs better than *PPCC* with relatively large margin, especially in the ACROSS protocol. Since in the ACROSS protocol, queries are searched among tracklets from all movies, the connections based on body appearance are not reliable across movies as those in the IN protocol. Thus by updating the gates between tracklets during inference, UGG is able to achieve much better performance than PPCC which is based on a fixed graph.

3. Supervised vs Unsupervised: From *UGG-U(favg)* to *UGG-T(favg)*, we observe significant improvements brought by training. It demonstrates that with labeled data, the UGG module can be inserted into deep networks for end-to-end training and achieve further performance improvement.

4. Semi-Supervised vs Unsupervised: We observe considerable improvements from *UGG-U(favg)* to *UGG-ST(favg)*. It implies that by reliable information

propagation in the graphs, the UGG module can be trained with only partially-labeled data, and still achieves results comparable to the supervised setting.

Methods	CSM		IJB-S	
	IN	ACROSS	S2SG	S2B
PPCC [46]	2.23s	458.56s	571.31s	580.16s
UGG-U	2.60s	41.85s	104.88s	111.35s

Table 5.4: Average run time on CSM and IJB-S datasets.

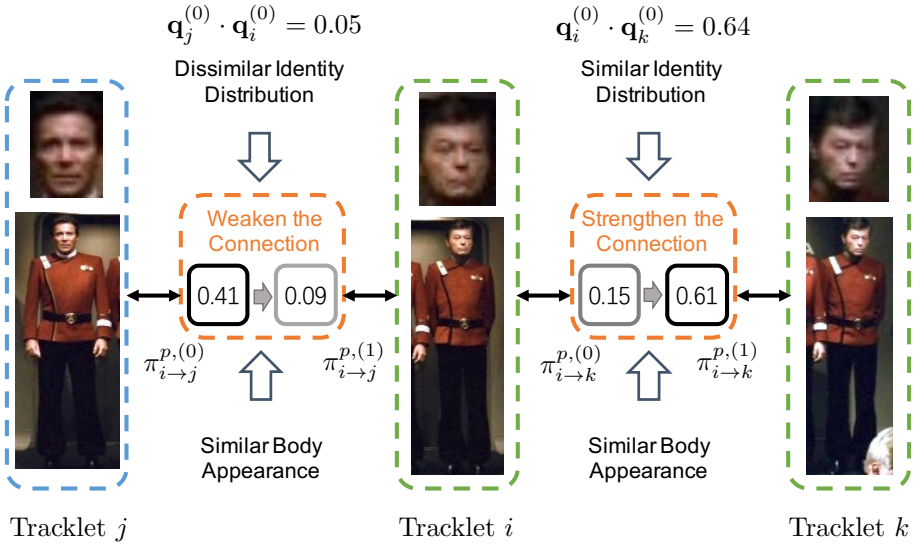


Figure 5.4: A qualitative example from the CSM dataset. The positive connection between tracklets i and j is initially strong because of the similar body appearance. During the inference step of the proposed method, this connection is weakened because of the divergent identity distributions between the two tracklets. It avoids erroneous information propagation through the connection. In contrast, the connection between tracklets i and k is strengthened due to their similar identity distributions.

5.4.4.2 Observations on IJB-S

1. UGG vs FACE and PPCC [46]: *UGG-U* performs better than *FACE* and *PPCC* on almost all evaluation metrics with relatively large margin, in both protocols, which again shows the effectiveness of the proposed method.

2. UGG + Better Similarity Metric: *UGG-U(sub)* achieves state-of-the-art results by combining subspace-based similarity and UGG. It shows that the proposed method can further improve the performance over the improvement from the similarity metric.

3. EERR Metric: EERR metric [52] is relatively lower than identification accuracy, because it penalizes missed face detections, which is out of the scope of this method.

Run time: From Table 5.4, we observe that UGG runs five times faster than PPCC on most of the protocols, which shows that UGG is more suitable for testing on large graphs during inference.

Qualitative Results: To illustrate the effectiveness of the proposed approach, a qualitative example is also shown in Figure 5.4. Tracklets i and j belong to different identities and tracklets i and k belong to the same identity. The initialized positive gate probability $\pi_{i\to j}^{p,(0)} = 0.41$ is greater than $\pi_{i\to k}^{p,(0)} = 0.15$. If the gate is fixed, information will be erroneously propagated between i and j. Using the proposed method, we can adaptively update the gate based on the identity information from i and j. Since identity distribution similarity $\mathbf{q}_j^{(0)} \cdot \mathbf{q}_i^{(0)} = 0.05$ is very small, the two tracklets are unlikely to have the same identity. Hence the positive connection

$\pi_{i\to j}^{p,(1)} = 0.09$ is weakened after the update. Similarly, since $\mathbf{q}_i^{(0)} \cdot \mathbf{q}_k^{(0)} = 0.64$ is large, the positive connection $\pi_{i\to k}^{p,(1)} = 0.61$ is strengthened correspondingly.

Configurations				CSM in avg				CSM in max				IJB-S in favg			
				IN		ACROSS		IN		ACROSS		S2SG		S2B	
PG	PGcl	NG	aG	mAP	R@1	mAP	R@1	mAP	R@1	mAP	R@1	A@1	E@1	A@1	E@1
				58.72%	76.19%	55.67%	53.15%	61.29%	76.64%	58.20%	54.60%	64.86%	29.62%	66.48%	30.38%
✓				61.14%	84.95%	62.00%	66.02%	61.60%	84.79%	62.05%	64.63%	71.21%	30.66%	72.05%	31.37%
	✓			-	-	-	-	-	-	-	-	71.26%	30.73%	72.16%	31.54%
	✓	✓		-	-	-	-	-	-	-	-	73.24%	32.35%	73.78%	32.88%
✓			✓	**62.81%**	**85.21%**	**63.30%**	**66.73%**	**63.74%**	**84.93%**	**63.42%**	**65.72%**	72.32%	30.92%	73.15%	31.64%
	✓		✓	-	-	-	-	-	-	-	-	72.46%	31.02%	73.28%	31.73%
	✓	✓	✓	-	-	-	-	-	-	-	-	**74.20%**	**32.70%**	**74.79%**	**33.29%**

Table 5.5: Ablation study. In configurations, *PG* stands for adding positive gates for positive information. *PGcl* stands for adding positive gates with extra control from cannot-links. *NG* stands for adding negative gates for negative information. *aG* stands for adaptively updating positive gates. *A@1* stands for Average Accuracy *with filtering* at R@1. *E@1* stands for EERR *without filtering* at R@1.

5.4.5 Ablation Studies

We conduct ablation studies on CSM and IJB-S datasets to show the effectiveness of key features in the proposed model. The results are shown in Table 5.5. We start from the baseline *FACE* without any information propagation, then gradually add key features of the method: *PG:* add fixed positive gates to propagate positive information. *PGcl:* same as *PG* except that positive information will not be propagated when cannot-link exists. *NG:* add negative gates to propagate negative

Configurations			IN				ACROSS			
PGTrain	aGTrain	UGGTest	mAP	R@1	R@3	R@5	mAP	R@1	R@3	R@5
			61.13%	77.86%	91.79%	96.65%	58.34%	56.56%	63.83%	66.34%
✓			61.39%	77.99%	91.77%	96.61%	58.94%	57.31%	64.26%	66.88%
✓	✓		61.40%	78.12%	91.85%	96.67%	58.70%	57.64%	64.49%	67.22%
		✓	64.14%	85.90%	95.42%	98.10%	65.82%	69.45%	76.83%	79.34%
✓		✓	64.58%	86.36%	95.53%	**98.27%**	66.90%	70.74%	77.83%	80.02%
✓	✓	✓	**64.60%**	**86.68%**	**95.56%**	98.24%	**67.09%**	**71.31%**	**77.93%**	**80.39%**

Table 5.6: Additional study on semi-supervised training on CSM dataset. *PGTrain* stands for using fixed positive gates during training. *aGTrain* stands for adaptively updating the gates during training. *UGGTest* stands for using UGG model during testing. In all experiments, only 25% of the training samples are labeled.

information. *aG:* adaptively update positive gates in *PG* or *PGcl* using the proposed method. Since detection information is not given in the CSM dataset, there is no co-occurrence cannot-links available and we do not use negative gates in this dataset. Thus, the proposed method *UGG-U* corresponds to *PG*+*aG* on the CSM dataset and *PGcl*+*NG*+*aG* on the IJB-S dataset.

From Table 5.5 we observe that: **1)** by introducing fixed positive gates, the performance improves compared to the baseline results, which indicates that positive information propagation controlled by body similarity improves the performance. **2)** by adding cannot-links to control the positive gates as well, marginal improvements are obtained. Thus, the performance improvement is limited if allow only positive information to propagate. **3)** by introducing additional negative gates using the same cannot-links, the performance improves significantly, which demonstrates the

effectiveness of allowing negative information to propagate between tracklets. **4)** finally, by adaptively updating the positive gates, we achieve the best performance in all protocols of both datasets. The result implies the advantages of adaptively updated gates.

5.4.6 Experiments on Different Training Settings

We also perform additional experiments on semi-supervised training on the CSM dataset with results shown in Table 5.6. We basically follow the regular training settings on the CSM datasets. The differences are

- For each movie, similar to the *UGG-ST* setting, we use all the galleries and randomly pick about 1/4 of the tracklets to construct the graph. Then we randomly pick 25% tracklets in the graph as labeled samples, and the rest 75% as unlabeled samples.
- We only train the 256×256 linear embedding matrix $\mathbf{W}_F^{gt}$ on the face features. Other embeddings are fixed as identity matrices.
- During training, we fix the parameters as $T_{gt} = 10$, $T_{tt} = 15$, $\alpha_p = 5$, $K = 2$, $\lambda = 0$ and $\lambda_f = 0.1$. $\lambda = 0$ because we are not training the pairwise embeddings.

Suppose after applying the embedding we want to learn, the similarities between galleries and labeled/unlabeled tracklets are $\mathbf{S}^{gt} = \left[\mathbf{S}_l^{gt}, \mathbf{S}_u^{gt}\right]$. We use three different settings to train the embedding: **1)** directly train on the labeled similarities

$\mathbf{S}_l^{gt}$ using cross-entropy loss, without invoking the UGG module. **2)** use the UGG module with positive gates to process $\mathbf{S}^{gt}$ and train on the output similarity $\tilde{\mathbf{S}}_l^{gt}$ corresponding to the labeled tracklets by cross-entropy loss, denoted as *PGTrain*. **3)** adaptively update the positive gates used in *PGTrain*, denoted as *aGTrain*.

Two settings are used to test the performance of the embedding: **1)** directly test on $\mathbf{S}^{gt}$ from the learned embedding, without using the UGG as post-processing. **2)** test on $\tilde{\mathbf{S}}^{gt}$ from the learned embedding and with the UGG post-processing, denoted as *UGGTest*.

From the results in Table 5.6, we observe that in the semi-supervised setting, the embedding trained with the UGG is more discriminative than the one trained without the module. It achieves better performance in both test settings. It shows that by propagating information between tracklets, the UGG also leverages the information from those unlabeled tracklets during training, which is important for semi-supervised learning. Also, the UGG with adaptive gates performs better than fixed gates, which demonstrates that adaptive gates is also helpful during training by propagating the information more precisely between tracklets.

5.5 Concluding Remarks

In this chapter, we proposed a graphical model-based method for video-based face recognition. The method propagates positive and negative identity information between tracklets through adaptive connections, which are influenced by both contextual information and identity distributions between tracklets. The proposed

method can be either used for post-processing, or trained in supervised and semi-supervised fashions. It achieves state-of-the-art results on CSM and IJB-S datasets. Ablation study further implies the effectiveness of the key features of the proposed method.

Chapter 6: Conclusions and Directions for Future Research

6.1 Summary

In this dissertation, we studied the problem of unconstrained still/video-based face recognition using augmented deep representations. In Chapter 2, we proposed two deep representation augmentation methods FV-DCNN and VLAD-DCNN to tackle the large pose variations in unconstrained face recognition by leveraging spatial information. We demonstrated the effectiveness of FV-DCNN on LFW and CFP datasets and showed that the FV-DCNN method can capture both local and global variations in the convolutional feature maps. The experiments on the challenging JANUS dataset show the effectiveness of VLAD-DCNN. We also compared the performance of VLAD-DCNN and FV-DCNN on the JANUS datasets and concluded that VLAD encoding works better than FV encoding because of the noisy second order statistics used by FV.

In Chapter 3, we proposed an automatic face recognition system for unconstrained video-based face recognition. The proposed system consists of modules for face detection and alignment, face association and tracking, face representation, subspace learning and matching. In the last module, we used subspaces to utilize the correlation between deep representations in video face sets. These subspaces along

with quality-aware exemplars of templates are used to produce the similarity scores between video pairs by a quality-aware principal angle-based subspace-to-subspace similarity metric. We evaluated the system on four video datasets. The experimental results demonstrated the superior performance of the proposed system and the subspace learning and matching method.

In Chapter 4, we proposed a dictionary learning and matching approach using deep representations for unconstrained video-based face verification. The proposed method learns structural and dynamical dictionaries from faces in video frames to incorporate temporal information and help improve the face recognition performance. Using the proposed LDDL algorithm, dynamical dictionaries and LDSs are jointly learned from the videos. A subspace-to-subspace similarity metric is defined for comparisons between videos. We evaluated the approach on three video datasets. The experiment results demonstrated the effectiveness of the proposed approach.

In Chapter 5, we proposed the UGG model for unconstrained video-based face recognition. The framework propagate identity information (computed from deep representations) from high-quality samples to low-quality samples through connections derived from contextual information in videos. It uses gate nodes and the associated random fields to model the uncertainty of connections between samples and enable the edge weights to be adaptively adjusted according to the neighboring samples. The gates also allow both positive and negative information to propagate simultaneously. The proposed method can be either used for post-processing, or trained in supervised and semi-supervised fashion. It achieves state-of-the-art results on CSM and IJB-S datasets which validates the effectiveness of the proposed

method.

6.2 Directions for Future Research

In this section, we outline several promising future directions that could be explored.

In Chapter 2, we studied the augmentation of deep representations for face recognition. A possible extension would be to design an supervised end-to-end FV-DCNN or VLAD-DCNN network for face recognition. Currently, the DCNN model is fixed when we are learning the FV/VLAD representation. Also, the Gaussians in FV encoding or the means in VLAD encoding are learned in an unsupervised manner. In order to achieve better performance for the face recognition problem, if sufficient amount of labeled training data is available, we can jointly finetune the DCNN model and the FV/VLAD model in a supervised way. Some work along these lines has been done in [3]. Also, since features from different layers of the network contain different levels of local information, an end-to-end multi-layer VLAD-DCNN model utilizing features from different layers can be developed. The analysis of VLAD and FV encoding on features from other DCNN architectures can also be pursued leading to new feature encoding techniques.

In Chapter 3, we used subspace representation to model the correlation between deep representations in a video. To improve the discriminative power of subspace representation, we could replace subspace-to-subspace similarity by manifold-to-manifold similarity. Since videos are not always single shot, faces in different parts

of the video may have different pose, illumination, expression and quality. Therefore a video partitioning method can be applied to separate different shots from a video. Instead of a single subspace, each video can be modeled by a manifold consisting of several component subspaces, where each component subspace is learned from a single shot. Then the manifold-to-manifold distance can be computed similar to the method proposed in [109].

In Chapter 4, we exploited the temporal information in videos by a linear dynamical dictionary learning method for video-based face recognition. A future extension of this method is to use the transition matrices learned by LDDL for other tasks like video-based facial expression recognition. Compared to dictionaries that capture the appearance information like the shape of mouths, eyes and eyebrows, transition matrices can capture the motion information in the video such as the opening of mouths, eyes, and the changing of eyebrow shapes, which is also discriminative for facial expressions.

We studied how a graphical-model assists video-based face recognition in Chapter 5. Currently we only use body appearance similarity and spatio-temporal positions of detections as contextual information. An interesting future work will be using reliable attribute information, such as gender, to construct negative connections and adaptively update negative gates. Another extension would be to make the method taking video streams as input. Instead of processing the whole video and build the graph at a time, the method will build the graph incrementally as it receives frames from the video stream. Face recognition can also be performed jointly with face association and tracking, in order to improve the overall performance.

Bibliography

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

[2] M. Aharon, M. Elad, and A. M. Bruckstein. The k-svd: an algorithm for designing of overcomplete dictionaries for sparse representation. *IEEE Transaction on Signal Process.*, 54(11), 2006.

[3] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[4] A. Bansal, A. Nanduri, C. D. Castillo, R. Ranjan, and R. Chellappa. Umdfaces: An annotated face dataset for training deep networks. *IEEE International Joint Conference on Biometrics (IJCB)*, 2017.

[5] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *ICIP*, pages 3464–3468, 2016.

[6] N. Bodla, J. Zheng, H. Xu, J.-C. Chen, C. Castillo, and R. Chellappa. Deep heterogeneous feature fusion for template-based face recognition. In *WACV*, pages 586–595, 03 2017.

[7] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*, 2018.

[8] Q. Cao, Y. Ying, and P. Li. Similarity metric learning for face recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2408–2415, 2013.

[9] 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.

[10] C.-H. Chen, J.-C. Chen, C. D. Castillo, and R. Chellappa. Video-based face association and identification. *12th FG*, pages 149–156, 2017.

[11] 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.

[12] 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.

[13] J.-C. Chen, W.-A. Lin, J. Zheng, and R. Chellappa. A real-time multi-task single shot face detector. *ICIP*, 2018.

[14] J. C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep CNN features. In *WACV*, March 2016.

[15] J.-C. Chen, R. Ranjan, S. Sankaranarayanan, A. Kumar, C.-H. Chen, V. M. Patel, C. D. Castillo, and R. Chellappa. Unconstrained still/video-based face verification with deep convolutional neural networks. *IJCV*, 126(2):272–291, Apr 2018.

[16] 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.

[17] J.-C. Chen, J. Zheng, V. M. Patel, and R. Chellappa. Fisher vector encoded deep convolutional features for unconstrained face verification. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2981–2985, Sep. 2016.

[18] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *TPAMI*, 40(4):834–848, 2018.

[19] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *CoRR*, abs/1412.7062, 2015.

[20] Y.-C. Chen, V. M. Patel, P. J. Phillips, and R. Chellappa. Dictionary-based face recognition from video. *ECCV*, October 2012.

[21] Y. C. Chen, V. M. Patel, P. J. Phillips, and R. Chellappa. Dictionary-based face and person recognition from unconstrained video. *IEEE Access*, 3:1783–1798, 2015.

[22] M. Cimpoi, S. Maji, I. Kokkinos, and A. Vedaldi. Deep filter banks for texture recognition, description, and segmentation. *International Journal of Computer Vision*, pages 1–30, 2016.

[23] N. Crosswhite, J. Byrne, O. M. Parkhi, C. Stauffer, Q. Cao, and A. Zisserman. Template adaptation for face verification and identification. *CoRR*, abs/1603.03958, 2016.

[24] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[25] J. Deng, J. Guo, X. Niannan, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

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

[27] C. Ding and D. Tao. Trunk-branch ensemble convolutional neural networks for video-based face recognition. *CoRR*, abs/1607.05427, 2016.

[28] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[29] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *IJCV*, 51:91–109, 2003.

[30] M. Du and R. Chellappa. Face association across unconstrained video frames using conditional random fields. In *Computer Vision – ECCV 2012*, pages 167–180, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[31] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, Apr. 1999.

[32] M. Elad. *Sparse and Redundant Representations: From theory to applications in Signal and Image processing*. Springer, 2010.

[33] B. Ghanem and N. Ahuja. Sparse coding of linear dynamical systems with an application to dynamic texture recognition. In *ICPR*, 2010.

[34] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. Detectron. `https://github.com/facebookresearch/detectron`, 2018.

[35] S. Gong, Y. Shi, A. K. Jain, and N. D. Kalka. Recurrent embedding aggregation network for video face recognition. *CoRR*, abs/1904.12019, 2019.

[36] S. Gong, Y. Shi, N. D. Kalka, and A. K. Jain. Video face recognition: Component-wise feature aggregation network (C-FAN). *CoRR*, abs/1902.07327, 2019.

[37] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. *Multi-scale Orderless Pooling of Deep Convolutional Activation Features*, pages 392–407. Springer International Publishing, Cham, 2014.

[38] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In *ECCV*, 2016.

[39] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017.

[40] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *arXiv prepring arXiv:1506.01497*, 2015.

[41] 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.

[42] M. Henaff, J. Bruna, and Y. LeCun. Deep convolutional networks on graph-structured data. *CoRR*, abs/1506.05163, 2015.

[43] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[44] H. Hu, G.-T. Zhou, Z. Deng, Z. Liao, and G. Mori. Learning structured inference neural networks with label relations. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2960–2968, 2016.

[45] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, University of Massachusetts, Amherst, 2007.

[46] Q. Huang, W. Liu, and D. Lin. Person search in videos with one portrait through visual and temporal links. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, pages 437–454, Cham, 2018. Springer International Publishing.

[47] W. Huang, F. Sun, L. Cao, D. Zhao, H. Liu, and M. Harandi. Sparse coding and dictionary learning with linear dynamical systems. In *CVPR*, June 2016.

[48] H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3304–3311, 2010.

[49] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(9):1704–1716, Sept. 2012.

[50] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[51] Z. Jiang, Z. Lin, and L. S. Davis. Label consistent k-svd: Learning a discriminative dictionary for recognition. *TPAMI*, 35(11), Nov 2013.

[52] N. D. Kalka, B. Maze, J. A. Duncan, K. J. OConnor, S. Elliott, K. Hebert, J. Bryan, and A. K. Jain. IJB-S : IARPA Janus Surveillance Video Benchmark. 2018.

[53] 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.

[54] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[55] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In *CVPR*, June 2015.

[56] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 109–117. Curran Associates, Inc., 2011.

[57] 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.

[58] V. Kumar, A. M. Namboodiri, and C. V. Jawahar. Face recognition in videos by label propagation. In *2014 22nd International Conference on Pattern Recognition*, pages 303–308, Aug 2014.

[59] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[60] C.-W. Lee, W. Fang, C.-K. Yeh, and Y.-C. F. Wang. Multi-label zero-shot learning with structured knowledge graphs. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1576–1585, 2018.

[61] Y. Liu, Y. Junjie, and W. Ouyang. Quality aware network for set to set recognition. In *CVPR*, 2017.

[62] B. Lu, R. Chellappa, and N. M. Nasrabadi. Incremental dictionary learning for unsupervised domain adaptation. In *BMVC*, 2015.

[63] B. Lu, J. Zheng, J. Chen, and R. Chellappa. Pose-robust face verification by exploiting competing tasks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1124–1132, March 2017.

[64] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *TPAMI*, 34(4), 2012.

[65] I. Masi, A. T. Trn, T. Hassner, J. T. Leksut, and G. Medioni. *Do We Really Need to Collect Millions of Faces for Effective Face Recognition?* 2016.

[66] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, and P. Grother. IARPA Janus Benchmark - C: Face dataset and protocol. In *2018 International Conference on Biometrics (ICB)*, pages 158–165, Feb 2018.

[67] Information Technology Laboratory, NIST. Multiple biomertic grand challenge, http : //www.nist.gov/itl/iad/ig/mbgc.cfm.

[68] M. Najibi, P. Samangouei, R. Chellappa, and L. Davis. SSH: Single stage headless face detector. In *ICCV*, 2017.

[69] A. J. O'Toole, J. Harms, S. L. Snow, D. R. Hurst, M. R. Pappas, J. H. Ayyad, and H. Abdi. A video database of moving faces and people. *TPAMI*, 27(5), May 2005.

[70] A. J. OToole, P. J. Phillips, S. Weimer, D. A. Roark, J. Ayyad, R. Barwick, and J. Dunlop. Recognizing people from dynamic and static faces and bodies: Dissecting identity with a fusion approach. *Vision Research*, 51(1), 2011.

[71] 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.

[72] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, 2015.

[73] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[74] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 40–44, 1993.

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

[76] J. K. Pillai, V. M. Patel, R. Chellappa, and N. Ratha. Secure and robust iris recognition using random projections and sparse representations. *TPAMI*, 33(9), Sept. 2011.

[77] Q. Qiu, V. M. Patel, and R. Chellappa. Information-theoretic dictionary learning for image classification. *TPAMI*, 36(11), 2014.

[78] R. Ranjan, A. Bansal, H. Xu, S. Sankaranarayanan, J. Chen, C. D. Castillo, and R. Chellappa. Crystal loss and quality pooling for unconstrained face verification and recognition. *CoRR*, abs/1804.01159, 2018.

[79] R. Ranjan, A. Bansal, J. Zheng, H. Xu, J. Gleason, B. Lu, A. Nanduri, J.-C. Chen, C. Castillo, and R. Chellappa. A fast and accurate system for face detection, identification, and verification. *CoRR*, abs/1809.07586, 2018.

[80] 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. *TPAMI*, pages 1–1, 2017.

[81] R. Ranjan, S. Sankaranarayanan, A. Bansal, N. Bodla, J.-C. Chen, V. M. Patel, C. D. Castillo, and R. Chellappa. Deep learning for understanding faces: Machines may be just as good, or better, than humans. *IEEE Signal Processing Magazine*, 35:66–83, 2018.

[82] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa. An all-in-one convolutional neural network for face analysis. In *12th IEEE FG*, volume 00, pages 17–24, May 2017.

[83] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*. 2015.

[84] R. Rubinstein, A. M. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6), june 2010.

[85] P. Saisan, G. Doretto, Y. Wu, and S. Soatto. Dynamic texture recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 58–63, 2001.

[86] S. Sankaranarayanan, A. Alavi, C. D. Castillo, and R. Chellappa. Triplet probabilistic embedding for face verification and clustering. *CoRR*, abs/1604.05417, 2016.

[87] S. Sankaranarayanan, A. Alavi, and R. Chellappa. Triplet similarity embedding for face verification. *CoRR*, abs/1602.03418, 2016.

[88] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, Jan 2009.

[89] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, June 2015.

[90] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. Jacobs. Frontal to profile face verification in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, 2016.

[91] R. Sheikh, M. Garbade, and J. Gall. Real-time semantic segmentation with label propagation. In *ECCV Workshops*, 2016.

[92] Y. Shen, H. Li, S. Yi, D. Chen, and X. Wang. Person re-identification with deep similarity-guided graph neural network. In *ECCV*, 2018.

[93] Y. Shi, A. K. Jain, and N. D. Kalka. Probabilistic face embeddings. *CoRR*, abs/1904.09658, 2019.

[94] 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.

[95] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[96] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *NIPS*. 2014.

[97] Y. Sun, D. Liang, X. Wang, and X. Tang. Deepid3: Face recognition with very deep neural networks. *CoRR*, abs/1502.00873, 2015.

[98] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *CVPR*, June 2015.

[99] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017.

[100] 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.

[101] M. Sznaier. Compressive information extraction: A dynamical systems approach. *System Identification*, 16:1559–1568, 2012.

[102] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

[103] S. Tripathi, S. J. Belongie, Y. Hwang, and T. Q. Nguyen. Detecting temporally consistent objects in videos through object class label propagation. *2016 IEEE*

Winter Conference on Applications of Computer Vision (WACV), pages 1–9, 2016.

[104] P. Turaga, A. Veeraraghavan, and R. Chellappa. Unsupervised view and rate invariant clustering of video sequences. *Computer Vision and Image Understanding*, 113(3):353–371, 2009.

[105] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

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

[107] R. Wang and X. Chen. Manifold discriminant analysis. In *CVPRW*, pages 429–436, 06 2009.

[108] R. Wang, H. Guo, L. S. Davis, and Q. Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *CVPR*, pages 2496–2503, 2012.

[109] R. Wang, S. Shan, X. Chen, Q. Dai, and W. Gao. Manifold-manifold distance and its application to face recognition with image sets. *IEEE TIP*, 21(10), 2012.

[110] X. Wang, Y. Ye, and A. Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. *CVPR*, 2018.

[111] X. Wei, H. Shen, and M. Kleinsteuber. An adaptive dictionary learning approach for modeling dynamical textures. In *ICASSP*, pages 3567–3571, May 2014.

[112] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, and K. Allen. IARPA Janus Benchmark-B face dataset. *CVPRW*, 2017.

[113] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. *CVPR*, pages 529–534, 2011.

[114] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6):1031–1044, 2010.

[115] J. Wright, A. Y. Yang, A. A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *TPAMI*, 31(2), feb. 2009.

[116] W. Xie and A. Zisserman. Multicolumn networks for face recognition. *ArXiv*, abs/1807.09192, 2018.

[117] H. Xu, J. Zheng, A. Alavi, and R. Chellappa. Learning a structured dictionary for video-based face recognition. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, March 2016.

[118] H. Xu, J. Zheng, A. Alavi, and R. Chellappa. Template regularized sparse coding for face verification. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 1448–1454, Dec 2016.

[119] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation network for video face recognition. In *CVPR*, pages 4362–4371, 2017.

[120] M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. In *ICCV*, 2011.

[121] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *CoRR*, abs/1411.7923, 2014.

[122] M. Zeiler, D. Krishnan, G. Taylor, and R. Fergus. Deconvolutional networks. In *CVPR*, 2010.

[123] Q. Zhang and B. Li. Discriminative k-svd for dictionary learning in face recognition. In *CVPR*, 2010.

[124] J. Zheng, J.-C. Chen, N. Bodla, V. M. Patel, and R. Chellappa. Vlad encoded deep convolutional features for unconstrained face verification. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 4101–4106, Dec 2016.

[125] J. Zheng, J.-C. Chen, V. M. Patel, C. D. Castillo, and R. Chellappa. Hybrid dictionary learning and matching for video-based face verification. *BTAS*, 2019.

[126] J. Zheng, R. Ranjan, C.-H. Chen, J.-C. Chen, C. D. Castillo, and R. Chellappa. An automatic system for unconstrained video-based face recognition. *CoRR*, abs/1812.04058, 2018.

[127] J. Zheng, R. Yu, J.-C. Chen, B. Lu, C. D. Castillo, and R. Chellappa. Uncertainty modeling of contextual-connection between tracklets for unconstrained video-based face recognition. *ArXiv*, abs/1905.02756, 2019.

[128] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *Computer Vision, IEEE International Conference on*, 2015.

[129] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *International Conference on Computer Vision (ICCV)*, 2015.

[130] K. Zhou. Deep-person-reid. `https://github.com/KaiyangZhou/deep-person-reid`.

[131] X. Zhou, K. Yu, T. Zhang, and T. S. Huang. Image classification using supervector coding of local image descriptors. In *European Conference on Computer Vision*, pages 141–154, 2010.

[132] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, 2002.