An End-to-End System for Unconstrained Face Verification with Deep Convolutional Neural Networks

Jun-Cheng Chen[1], Rajeev Ranjan[1], Amit Kumar[1], Ching-Hui Chen[1], Vishal M. Patel[2], and Rama Chellappa[1]
1. University of Maryland, College Park
2. Rutgers, The State University of New Jersey
pullpull@cs.umd.edu, rranjan1@umd.edu, {iitkgp.ece.amit,ezhuei}@gmail.com,
vishal.m.patel@rutgers.edu, Rama@umiacs.umd.edu

Abstract

In this paper, we present an end-to-end system for the unconstrained face verification problem based on deep convolutional neural networks (DCNN). The end-to-end system consists of three modules for face detection, alignment and verification and is evaluated using the newly released IARPA Janus Benchmark A (IJB-A) dataset and its extended version Janus Challenging set 2 (JANUS CS2) dataset. The IJB-A and CS2 datasets include real-world unconstrained faces of 500 subjects with significant pose and illumination variations which are much harder than the Labeled Faces in the Wild (LFW) and Youtube Face (YTF) datasets. Results of experimental evaluations for the proposed system on the IJB-A dataset are provided.

1. Introduction

Face verification is one of the core problems in computer vision and has been actively researched for over two decades [44]. In face verification, given two videos or images, the objective is to determine whether they belong to the same person. Many algorithms have been shown to work well on images that are collected in controlled settings. However, the performance of these algorithms often degrades significantly on images that have large variations in pose, illumination, expression, aging, and occlusion. Most face verification systems assume that the faces have already been detected and focus on designing matching algorithms. However, for an automatic face verification system to be effective, it needs to handle errors that are introduced by algorithms for automatic face detection, face association, and facial landmark detection algorithms.

Existing methods have focused on learning robust and discriminative representations from face images and videos. One approach is to extract an over-complete and high-dimensional feature representation followed by a learned metric to project the feature vector into a low-dimensional space and to compute the similarity scores. For example, high-dimensional multi-scale local binary pattern (LBP) [8] features extracted from local patches around facial landmarks and Fisher vector (FV) [33][10] features have been shown to be effective for face recognition. However, deep convolutional neural networks (DCNN) have demonstrated impressive performances on different tasks such as object recognition [26][37], object detection [21][27], and face verification [31]. It has been shown that a deep convolutional neural network model can not only characterize large data variations but also learn a compact and discriminative feature representation when the size of the training data is sufficiently large. In addition, it can be easily generalized to other vision tasking by fine-tuning the pretrained model on the new task [18].

In this work, we present an end-to-end automatic face verification system. Due to the robustness of DCNN features, we also build the face preprocessing modules (*i.e.* face detection, and facial landmark detection based on the same DCNN model used in [26].). We demonstrate that the face detection module performs much better than many commercial off-the-shelf software systems for the IJB-A dataset. For face verification, we train another DCNN model using a large-scale face dataset, the CASIA-WebFace [42]. Finally, we compare the performance of our method with one using manually annotated data and another using commercial off-the-shelf face matchers. This comparison is done on the challenging IJB-A dataset which contains significant variations in pose, illumination, expression, resolution and occlusion. The performance of our end-to-end system degrades only slightly as compared to the one that uses manually annotated data, (*i.e.* face bounding boxes and three facial landmark annotation, including left eye, right eye, and the nose base.), thus demonstrating the robustness of our system.

Although performance evaluations of many face match-

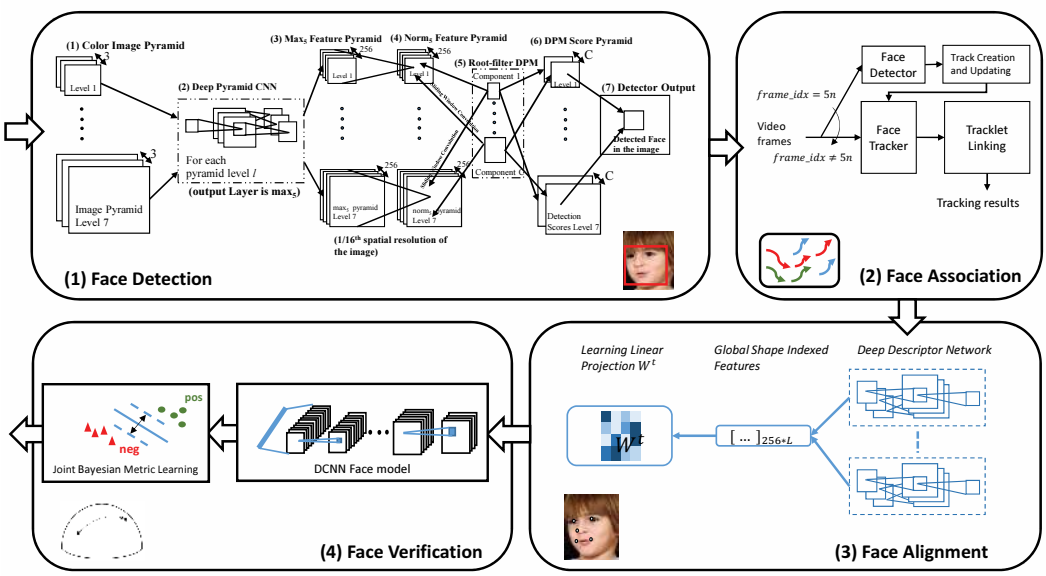


Figure 1. An overview of the proposed end-to-end DCNN-based face verification system.

ers on the IJB-A data set are being carried out or have been recently completed, these results are not publicly available yet. More importantly, at the time preparing this paper, these matchers have only been tested using manually annotated landmark points provided along with the IJB-A face data set. The proposed system is fully automatic and thus presents the performance of the end-to-end system for face verification using the IJB-A data and protocol.

The rest of the paper is organized as follows. We briefly review related works in Section 2. Details of the different components of the proposed end-to-end system, including face detection, face association, face alignment, and face verification, based on convolutional neural networks, are given in Section 3. Experimental results are presented in Section 4. Finally, we conclude the paper in Section 5 with a brief summary and discussion.

2. Related Works

A typical automated face verification system consists of the following components: (1) face detection and (2) face association to localize faces, (3) facial landmark detection to align faces, and (4) face verification to verify a subject's identity. Due to the large number of published papers in the literature, we briefly review only the relevant works for each component.

2.1. Face Preprocessing

Face Detection: The face detection method introduced by Viola and Jones [40] is based on the cascaded classifiers built using the Haar wavelet features to detect faces. Due to its simplicity, this method can work in real-time. Zhu *et al.* [45] improved the performance of the method using the deformable part model (DPM) framework, which treats each facial landmark as a part and uses the HOG features to simultaneously perform face detection, pose estimation, and landmark localization. However, the key challenge in unconstrained face detection is that features like Haar wavelets and HOG do not capture the salient facial information at different poses and illumination conditions. It has been shown in [18] that a deep CNN pre-trained with the Imagenet dataset can be used as a meaningful feature extractor for various vision tasks. Regions with CNN (R-CNN) [29] computes region-based deep features and attains state-of-art face detection performance. In addition, since the deep pyramid [22] removes the fixed-scale input dependency in deep CNNs, it makes it attractive to be integrated with the DPM and further improve the detection accuracy across scale [27].

Face Association: The video-based face verification system [11] requires consistently-tracked faces to capture the diverse pose and spatial-temporal information for analysis. In addition, there is usually more than one person shown in the videos, and thus multiple face images from different individuals should be correctly associated between video frames. Several recent techniques achieve multiple object tracking by modeling the motion context [43], track management [20], and guided tracking using the confidence map of the detector [4]. Roth *et al.* [30] proposed to adapt the framework of multiple object tracking to multiple face tracking based on tracklet linking, and several face-specific metrics and constraints have been introduced to enhance the

reliability of face tracking. A recent study [13] proposed managing the track from the continuous face detection output without relying on long-term observations. In unconstrained scenarios, the camera can be affected by abrupt movements, which poses consistent tracking challenging. Du *et al.* proposed a conditional random field (CRF) framework to associate faces in two consecutive frames by utilizing the affinity of facial features, location, motion, and clothing appearance [19].

Facial Landmark Detection: Facial landmark detection is an important component for a face verification system to align the faces into canonical coordinates and to improve the performance of the verification algorithms. Pioneering works such as Active Appearance Models (AAM) [14] and Active Shape Models (ASM) [15] are built using PCA constraints on appearance and shape. In [16], Cristinacce *et al.* generalized the ASM model to a Constrained Local Model (CLM), in which every landmark has a shape constrained descriptor to capture the appearance. Zhu *et al.* [45] used a part-based model for face detection, pose estimation and landmark localization assuming the face shape to be a tree structure. Asthana *et al.* [2] combined the discriminative response map fitting with CLM. In general, these methods learn a model that directly maps image appearance to the target output. Nevertheless, the performance of these methods depends on the robustness of local descriptors. As in [26], the deep features are shown to be robust to different challenging variations. Sun *et al.* [35] proposed a cascade of carefully designed CNNs in which at each level outputs of multiple networks are fused for landmark estimation and achieve good performance. Unlike [35], we use a single CNN, carefully designed to provide a unique key-point descriptor and achieve better performance.

2.2. Face Verification

Feature Learning: Learning invariant and discriminative feature representations is the first step for a face verification system. Ahonen *et al.* [1] showed that the Local Binary Pattern (LBP) is effective for face recognition. Chen *et al.* [8] demonstrated good results for face verification using the high-dimensional multi-scale LBP features extracted from patches around facial landmarks. However, recent advances in deep learning methods have shown that compact and discriminative representations can be learned using DCNN from very large datasets. Taigman *et al.* [39] learned a DCNN model on the frontalized faces generated with a general 3D shape model from a large-scale face dataset and achieved better performance than many traditional methods. Sun *et al.* [36] achieved results that surpass human performance for face verification on the LFW dataset using an ensemble of 25 simple DCNN with fewer layers trained on weakly aligned face images from a much smaller dataset than [39]. Schroff *et al.* [31] adapted the state-of-the-art deep architecture in object recognition to face recognition and trained on a large-scale unaligned private face dataset with the triplet loss. These studies essentially demonstrate the effectiveness of the DCNN model for feature learning and detection/recognition/verification problems.

Metric Learning: Learning a similarity measure from data is the other key component to improve the performance of a face verification system. Many approaches have been proposed in the literature that essentially exploit the label information from face images or face pairs. Taigman *et al.* [38] learned the Mahalanobis distance using the Information Theoretic Metric Learning (ITML) method [17]. Chen *et al.* [7] proposed a joint Bayesian approach for face verification which models the joint distribution of a pair of face images and uses the ratio of between-class and within-class probabilities as the similarity measure.

3. Proposed Approach

The proposed system is a complete pipeline for an automatic face verification system. We first perform face detection to localize faces in each image and video frame. Then, we associate the detected faces with the common identity across the videos and align the faces into canonical coordinates using the detected landmarks. Finally, we perform face verification to compute the similarity between a pair of images/videos. The system is illustrated in Figure 1. The details of each component are presented in the following sections.

3.1. Face Detection

All the faces in the images/video frames are detected with a DCNN-based face detector, called Deep Pyramid Deformable Parts Model for Face Detection (DP2MFD) [27], which consists of two modules. The first module generates a seven level normalized deep feature pyramid for any input image of arbitrary size, as illustrated in the first part of Figure 1 (*i.e.* the same CNN architecture as Alexnet[26] is adopted to extract the deep features). This image pyramid network generates a pyramid of 256 feature maps at the fifth convolution layer (conv5). A 3×3 max filter is applied to the feature pyramid at a stride of one to obtain the max5 layer. The activations at each level are further normalized in the (norm5) layer to remove the bias from face size. Fixed-length features from each location in the pyramid are extracted using the sliding window approach. The second module is a linear SVM, which takes these features as input to classify each location as face or non-face, based on their scores. In addition, the deep pyramid features are robust to not only pose and illumination variations but also to different scales. As shown in Figure 2, the DP2MFD algorithm works well in unconstrained settings. We present the face detection performance results under the face detection protocol of the IJB-A dataset in Section 4.

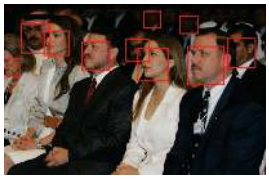


Figure 2. Sample detection results on an IJB-A image using the deep pyramid method.

3.2. Face Association

Because there are multiple subjects appearing in the frames of each video of the IJB-A dataset, performing face association to assign each face to its corresponding subject is an important step for us to pick the correct subject for face verification. Thus, once the faces in the images and video frames are detected, we perform multiple face tracking by integrating results from the face detector, face tracker, and tracklet linking. The second part of the Figure 1 shows the block diagram of the multiple face tracking system. We apply the face detection algorithm in every fifth frame using the face detection method presented in Section 3.1. A detected bounding box is considered as a novel detection if it does not have an overlap ratio with any bounding box in the previous frames larger than γ. The overlap ratio of a detected bounding box $\mathbf{b}_d$ and a bounding box $\mathbf{b}_{tr}$ in the previous frames is defined as

$$s(\mathbf{b}_d, \mathbf{b}_{tr}) = \frac{area(\mathbf{b}_d \cap \mathbf{b}_{tr})}{area(\mathbf{b}_{tr})}. \quad (1)$$

We empirically set the overlap threshold γ to 0.2. A face tracker is created from a detection bounding box that is treated as a novel detection. For face tracking, we use the Kanade-Lucas-Tomasi (KLT) feature tracker [32] to track the faces between two consecutive frames. To avoid the potential drifting of trackers, we update the bounding boxes of the tracker by those provided by the face detector in every five frames. The detection bounding box $\mathbf{b}_d$ replaces the tracking bounding boxes $\mathbf{b}_{tr}$ of a tracklet in the previous frame if $s(\mathbf{b}_d, \mathbf{b}_{tr}) \leq \gamma$. A face tracker is terminated if there is no corresponding face detection overlapping with it for more than t frames. We set t to 4 based on empirical grounds.

In order to handle the fragmented face tracks resulting from occlusions or unreliable face detection, we use the tracklet linking method proposed by [3] to associate the bounding boxes in the current frames with tracklets in the previous frames. The tracklet linking method consists of two stages. The first stage is to associate the bounding boxes provided by tracking or detection in current frames with the existing tracklet in previous frames. This stage consists of local and global associations. The local association step associates the bounding boxes with the set of tracklets, having high confidence. The global association step associates the remaining bounding boxes with the set of tracklets of low confidence. The second stage is to update the confidence of the tracklets, which will be used for determining the tracklets for local or global association in the first stage. We show sample face association results for videos in Figure 3.

Figure 3. Sample results of our face association method for videos.

Figure 4. Sample facial landmark detection results.

3.3. Facial Landmark Detection

Once the faces are detected, we perform facial landmark detection for face alignment. The proposed facial landmark detection algorithm works in two stages. We model the task as a regression problem, where beginning with the initial mean shape, the target shape is reached through regression. The first step is to perform feature extraction of a patch around a point of the shape followed by linear regression as described in [28][5]. Given a bounding box detected by the face detection method described in Section 3.1, we first initialize a mean shape over the face. The CNN features, carefully designed with the proper number of strides and pooling, are used as the features to perform regression. We use the same CNN architecture as Alexnet [26] with the pretrained weights for the ImageNet dataset. In addition, we finetune the CNN with the face detection task. This helps the network to learn features specific to faces. Furthermore, we adopt the cascade regression, in which the output generated by the first stage is used as an input for the next stage. We use 5 stages for our system. The patches selected for feature extraction are reduced subsequently in later stages to improve the localization of facial landmarks. After the facial landmark detection is completed, each face is aligned into the canonical coordinate with similarity transform using the 7 landmark points (i.e. two left eye corners, two right eye corners, nose tip, and two mouth corners). After alignment, the face image resolution is 100 $\times$ 100 pixels. Sample detected landmarks results are shown in Figure 4.

3.4. Face Verification

Deep Face Feature Representation: Stacking small filters to approximate large filters and building very deep convolutional networks reduces the number of parameters but also increases the nonlinearity of the network in [34][37]. In addition, the resulting feature representation is compact and discriminative. Therefore, we use the same network architecture presented in [9] and train it using the CASIA-WebFace dataset [42]. The dimensionality of the input layer is $100 \times 100 \times 1$ for gray-scale images. The network includes 10 convolutional layers, 5 pooling layers, and 1 fully connected layer. Each convolutional layer is followed by a parametric rectified linear unit (PReLU) [24], except the last one, Conv52. Moreover, two local normalization layers are added after Conv12 and Conv22, respectively, to mitigate the effect of illumination variations. The kernel size of all filters are 3×3. The first four pooling layers use the max operator, and pool_5 uses average pooling. The feature dimensionality of pool_5 is thus equal to the number of channels of Conv52 which is 320. The dropout ratio is set as 0.4 to regularize Fc6 due to the large number of parameters (*i.e.* 320×10548.). The pool_5 feature is used for face representation. The extracted features are further L_2-normalized to unit length before the metric learning stage. If there are multiple images and frames available for the subject template, we use the average of the pool_5 features as the overall feature representation.

Joint Bayesian Metric Learning: To leverage the positive and negative label information available from the training dataset, we learn a joint Bayesian metric which has demonstrated good performances on face verification problems [7][6]. It models the joint distribution of feature vectors of both ith and jth images, $\{\mathbf{x}_i, \mathbf{x}_j\}$, as a Gaussian. Let $P(\mathbf{x}_i, \mathbf{x}_j|H_I) \sim N(0, \mathbf{\Sigma}_I)$ when $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same class, and $P(\mathbf{x}_i, \mathbf{x}_j|H_E) \sim N(0, \mathbf{\Sigma}_E)$ when they are from different classes. Moreover, each face vector is modeled as, $\mathbf{x} = \boldsymbol{\mu} + \boldsymbol{\epsilon}$, where $\boldsymbol{\mu}$ is for the identity and $\boldsymbol{\epsilon}$ for pose, illumination, and other variations. Both $\boldsymbol{\mu}$ and $\boldsymbol{\epsilon}$ are assumed to be independent zero-mean Gaussian distributions, $N(0, \mathbf{S}_\mu)$ and $N(0, \mathbf{S}_\epsilon)$, respectively. The log likelihood ratio of intra- and inter-classes, $r(\mathbf{x}_i, \mathbf{x}_j)$, can be computed as follows:

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

where $\mathbf{M}$ and $\mathbf{R}$ are both negative semi-definite matrices. Equation (2) can be rewritten as $(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j) - 2\mathbf{x}_i^T \mathbf{B} \mathbf{x}_j$ where $\mathbf{B} = \mathbf{R} - \mathbf{M}$. More details can be found in [7]. Instead of using the EM algorithm to estimate $\mathbf{S}_\mu$ and $\mathbf{S}_\epsilon$, we use a large-margin framework to optimize the distance as follows:

$$\underset{\mathbf{M}, \mathbf{B}, b}{\operatorname{argmin}} \sum_{i,j} max[1 - y_{ij}(b - (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j) + 2\mathbf{x}_i^T \mathbf{B} \mathbf{x}_j), 0], \tag{3}$$

where $b \in \mathbb{R}$ 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. For simplicity, we denote $(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j) - 2\mathbf{x}_i^T \mathbf{B} \mathbf{x}_j$ as $d_{\mathbf{M},\mathbf{B}}(\mathbf{x}_i, \mathbf{x}_j)$. $\mathbf{M}$ and $\mathbf{B}$ are updated using stochastic gradient descent as follows and are equally trained on positive and negative pairs in turn:

$$\begin{aligned}
\mathbf{M}_{t+1} &= \begin{cases} \mathbf{M}_t, & \text{if } y_{ij}(b_t - d_{\mathbf{M},\mathbf{B}}(\mathbf{x}_i, \mathbf{x}_j)) > 1 \\ \mathbf{M}_t - \gamma y_{ij} \mathbf{\Gamma}_{ij}, & \text{otherwise,} \end{cases} \\
\mathbf{B}_{t+1} &= \begin{cases} \mathbf{B}_t, & \text{if } y_{ij}(b_t - d_{\mathbf{M},\mathbf{B}}(\mathbf{x}_i, \mathbf{x}_j)) > 1 \\ \mathbf{B}_t + 2\gamma y_{ij} \mathbf{x}_i \mathbf{x}_j^T, & \text{otherwise,} \end{cases} \\
b_{t+1} &= \begin{cases} b_t, & \text{if } y_{ij}(b_t - d_{\mathbf{M},\mathbf{B}}(\mathbf{x}_i, \mathbf{x}_j)) > 1 \\ b_t + \gamma_b y_{ij}, & \text{otherwise,} \end{cases}
\end{aligned} \tag{4}$$

where $\mathbf{\Gamma}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T$ and γ is the learning rate for $\mathbf{M}$ and $\mathbf{B}$, and γ_b for the bias b. We use random semi-definite matrices to initialize both $\mathbf{M} = \mathbf{V}\mathbf{V}^T$ and $\mathbf{B} = \mathbf{W}\mathbf{W}^T$ where both $\mathbf{V}$ and $\mathbf{W} \in \mathbb{R}^{d \times d}$, and v_{ij} and $w_{ij} \sim N(0, 1)$. In addition, $\mathbf{M}$ and $\mathbf{B}$ are updated only when the constraints are violated. In our implementation, the ratio of the positive and negative pairs that we generate based on the identity information of the training set is 1:20. Furthermore, the other reason to train the metric instead of using traditional EM is that for the IJB-A training and test data, some templates only contain a single image. More details about the IJB-A dataset are given in Section 4.

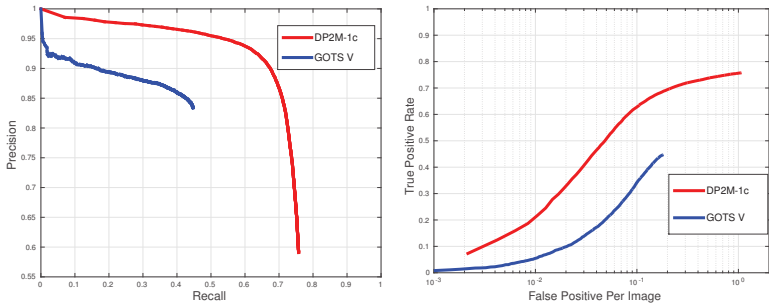
4. Experimental Results

In this section, we present the results of the proposed automatic system for both face detection and face verification tasks on the challenging IARPA Janus Benchmark A (IJB-A) [25], and its extended version Janus Challenging set 2 (JANUS CS2) dataset. The JANUS CS2 dataset contains not only the sampled frames and images in the IJB-A, but also the original videos. In addition, the JANUS CS2 dataset[1] includes considerably more test data for identification and verification problems in the defined protocols than the IJB-A dataset. The receiver operating characteristic curves (ROC) and the cumulative match characteristic (CMC) scores are used to evaluate the performance of different algorithms for face verification. The ROC curve measures the performance in the verification scenarios, and the CMC score measures the accuracy in a closed set identification scenarios.

4.1. Face Detection on IJB-A

The IJB-A dataset contains images and sampled video frames from 500 subjects collected from online media [25], [12]. For the face detection task, there are 67,183 faces of which 13,741 are from images and the remaining are from videos. The locations of all faces in the IJB-A dataset were manually annotated. The subjects were captured so that the dataset contains wide geographic distribution. Nine different face detection algorithms were evaluated on the IJB-A

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



(a) (b)

Figure 5. Face detection performance evaluation on the IJB-A dataset. (a) Precision vs. recall curves. (b) ROC curves.

dataset [12], and the algorithms compared in [12] include one commercial off the shelf (COTS) algorithm, three government off the shelf (GOTS) algorithms, two open source face detection algorithms (OpenCVs Viola Jones and the detector provided in the Dlib library), and PittPat ver 4 and 5. In Figure 5, we show the precision-recall (PR) curves and the ROC curves, respectively corresponding to the method used in our work and one of the best reported methods in [12]. From the results, the face detection algorithm used in our system outperforms the best performing method reported in [12] by a large margin.

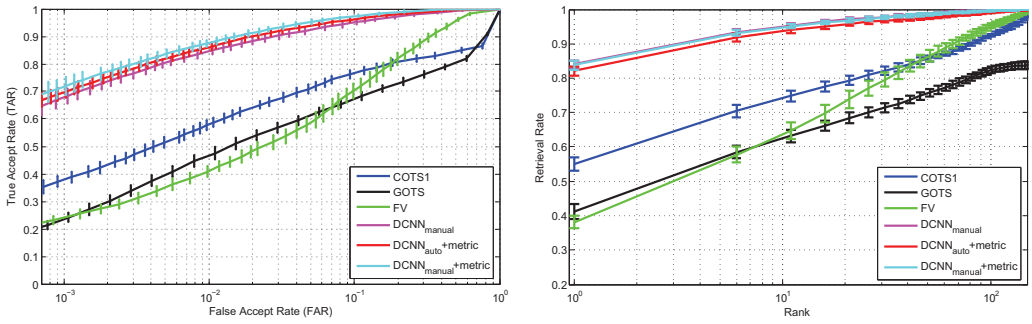
Figure 6. Sample images and frames from the IJB-A and JANUS CS2 datasets. Challenging variations due to pose, illumination, resolution, occlusion, and image quality are present in these images.

4.2. Face verification on IJB-A and JANUS CS2

For the face verification task, both IJB-A and JANUS CS2 datasets contain 500 subjects with 5,397 images and 2,042 videos split into 20,412 frames, 11.4 images and 4.2 videos per subject. Sample images and video frames from the datasets are shown in Figure 6. The videos are only released for the JANUS CS2 dataset. The IJB-A evaluation protocol consists of verification (1:1 matching) over 10 splits. Each split contains around 11,748 pairs of templates (1,756 positive and 9,992 negative pairs) on average. Similarly, the identification (1:N search) protocol also consists of 10 splits, which are used to evaluate the search performance. In each search split, there are about 112 gallery templates and 1763 probe templates (*i.e.* 1,187 genuine probe templates and 576 impostor probe templates). On the other hand, for the JANUS CS2, there are about 167 gallery templates and 1763 probe templates and all of them are used for both identification and verification. The training set for both datasets contains 333 subjects, and the test set contains 167 subjects. Ten random splits of training and testing are provided by each benchmark, respectively. The main differences between the IJB-A and JANUS CS2 evaluation protocols are that (1) IJB-A considers the open-set identification problem and the JANUS CS2 considers the closed-set identification and (2) IJB-A considers the more difficult pairs which are the subsets from the JANUS CS2 dataset. Both IJB-A and the JANUS CS2 datasets are divided into training and test sets. For the test sets of both benchmarks, the image and video frames of each subject are randomly split into gallery and probe sets without any overlapping subjects between them. Unlike the LFW and YTF datasets, which only use a sparse set of negative pairs to evaluate the verification performance, the IJB-A and JANUS CS2 both divide the images/video frames into gallery and probe sets so that all the available positive and negative pairs are used for the evaluation. Also, each gallery and probe set consist of multiple templates. Each template contains a combination of images or frames sampled from multiple image sets or videos of a subject. For example, the size of the similarity matrix for JANUS CS2 split1 is 167×1806 where 167 are for the gallery set and 1806 for the probe set (*i.e.* the same subject reappears multiple times in different probe templates). Moreover, some templates contain only one profile face with a challenging pose with low quality imagery. In contrast to the LFW and YTF datasets, which only include faces detected by the Viola Jones face detector [40], the images in the IJB-A and JANUS CS2 contain extreme pose, illumination, and expression variations. These factors essentially make the IJB-A and JANUS CS2 challenging face recognition datasets [25].

4.3. Evaluation on JANUS-CS2 and IJB-A

We compares the results generated by the proposed automatic system to those generated by the same DCNN model



(a) (b)

Figure 7. Results on the JANUS CS2 dataset. (a) the average ROC curves and (b) the average CMC curves.

IJB-A-Verif	[41]	DCNN$_{manual}$	DCNN$_{auto}$+metric	DCNN$_{manual}$+metric
FAR=1e-2	0.732±0.033	0.736±0.045	0.776±0.033	**0.787±0.043**
FAR=1e-1	0.895±0.013	0.91±0.015	0.936±0.01	**0.947±0.011**
IJB-A-Ident	[41]	DCNN$_{manual}$	DCNN$_{auto}$+metric	DCNN$_{manual}$+metric
Rank-1	0.820±0.024	**0.853±0.014**	0.834±0.017	0.852 ±0.018
Rank-5	0.929±0.013	**0.94±0.01**	0.922±0.011	0.937±0.01
Rank-10	N/A	**0.962±0.07**	0.947±0.011	0.954±0.007

Table 1. Results on the IJB-A dataset. The TAR of all the approaches at FAR=0.1 and 0.01 for the ROC curves. The Rank-1, Rank-5, and Rank-10 retrieval accuracies of the CMC curves.

JANUS-CS2-Verif	COTS1	GOTS	FV[33]	DCNN$_{manual}$	DCNN$_{auto}$+metric	DCNN$_{manual}$+metric
FAR=1e-2	0.581±0.054	0.467±0.066	0.411±0.081	0.847±0.016	0.861±0.014	**0.876±0.013**
FAR=1e-1	0.767±0.015	0.675±0.015	0.704±0.028	0.95±0.009	0.963±0.007	**0.973±0.005**
JANUS-CS2-Ident	COTS1	GOTS	FV [33]	DCNN$_{manual}$	DCNN$_{auto}$+metric	DCNN$_{manual}$+metric
Rank-1	0.551±0.03	0.413±0.022	0.381±0.018	**0.841±0.011**	0.82±0.014	0.838±0.012
Rank-5	0.694±0.017	0.571±0.017	0.559±0.021	**0.927±0.01**	0.91±0.01	0.924±0.009
Rank-10	0.741±0.017	0.624±0.018	0.637±0.025	**0.951±0.006**	0.938±0.01	0.949±0.006

Table 2. Results on the JANUS CS2 dataset. The TAR of all the approaches at FAR=0.1 and 0.01 for the ROC curves. The Rank-1, Rank-5, and Rank-10 retrieval accuracies of the CMC curves.

trained and tested using the manual annotated bounding boxes and facial landmarks of the dataset. We replace the video frames defined in the protocols for both JANUS CS2 and IJB-A of each template with our associated faces and use only one bounding box in the metadata to select the subject. For the images, we use our face detection boxes which have the largest overlapping boxes with the ones defined in the metadata for the images. We have made an effort to use the minimum possible metadata needed for a fair comparison. In addition, we also compare the results with the FV approach proposed in [33] and with two other commercial off-the-shelf matchers, COTS1 and GOTS [25] which are tested with the manually annotated data as well. The COTS1 and GOTS baselines provided by JANUS CS2 are the top performers from the most recent NIST FRVT study [23]. The FV method is trained on the LFW dataset which contains only a few faces with extreme poses. Therefore, we use the pose information estimated from the landmark detector and select face images/video frames whose yaw angles are less than or equal to ±25 degrees for each gallery and probe set. If there are no images/frames satisfying this constraint, we choose the one closest to the frontal one. However, for the DCNN method, we use all the frames without applying the selection strategy presented above.

We show the ROC and CMC scores generated by the proposed end-to-end system for the IJB-A dataset in Table 1 and compare it with the recent work [41] that uses the fused features from a 7-DCNN model. Figure 7 shows the ROC curves and the CMC curves for the JANUS CS2 dataset, respectively for verification and identification protocols. Their corresponding scores are summarized in Table 2. DCNN$_{manual}$ computes the similarity scores on the test data using cosine distance for the finetuned DCNN features where the pretrained DCNN model from the CASIA-WebFace dataset is further finetuned using the IJB-A and JANUS CS2 training data with manual annotation.. The distance is computed on test data with manual annotation. Metric stands for applying the learned metric to compute the

similarity. DCNN$_{auto}$ means performing all the finetuning, metric learning, and testing steps using the data processed with our own face preprocessing components. From the ROC and CMC curves, we see that the DCNN method performs better than other competitive methods. This can be attributed to the fact that the DCNN model does capture face variations over a large dataset and generalizes well to a new small dataset. In addition, the performance of the proposed automatic system degrades only slightly as compared to the one using the manual annotations. This demonstrates the robustness of each component of our system. After metric learning, we can see a slight performance degradation for the retrieval rate of the identification experiments. This may be due to the fact that the positive and negative pairs are uniformly and randomly selected. A quick solution would be to perform hard negative mining to make the learned metric ensure that the similarity scores for all the hard imposters to be small. We will consider this in the near future.

4.4. Run Time

The DCNN model for face verification is pretrained on the CASIA-Webface dataset for about 7 days using NVidia Titan X. The running time for face detection is around 0.7 second per image. The facial landmark detection and feature extraction steps take about 1 second and 0.006 second per face, respectively. The face association module for a video takes around 5 fps on average.

5. Conclusion

We presented the design and performance of our automatic face verification system, which automatically locates faces and performs verification/recognition on newly released challenging face verification datasets, IARPA Benchmark A (IJB-A) and its extended version, JANUS CS2. It was shown that our proposed DCNN-based system can not only accurately locate the faces across images and videos but also learn a robust model for face verification. Experimental results demonstrate that the performance of the proposed system on the IJB-A dataset is much better than a FV-based method and some COTS and GOTS matchers.

We plan to integrate the components of our face verification system into one system, which performs all of the tasks simultaneously with a multi-task formulation and train the deep convolutional neural network once instead of training each task separately.

6. Acknowledgments

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2014-14071600012. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. We thank professor Alice O'Toole for carefully reading the manuscript and suggesting improvements in the presentation of this work.

References

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006. 3

[2] A. Asthana, S. Zafeiriou, S. Y. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3444–3451, 2013. 3

[3] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 4

[4] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Robust tracking-by-detection using a detector confidence particle filter. In *IEEE International Conference on Computer Vision (ICCV)*, 2009. 2

[5] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression, July 3 2014. US Patent App. 13/728,584. 4

[6] 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. 5

[7] 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. 3, 5

[8] 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. 1, 3

[9] J.-C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep cnn features. *arXiv preprint arXiv:1508.01722*, 2015. 5

[10] 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. 1

[11] Y.-C. Chen, V. M. Patel, P. J. Phillips, and R. Chellappa. Dictionary-based face recognition from video. In *European Conference on Computer Vision (ECCV)*, 2012. 2

[12] J. Cheney, B. Klein, A. K. Jain, and B. F. Klare. Unconstrained face detection: State of the art baseline and challenges. In *International Conference on Biometrics*, 2015. 5, 6

[13] F. Comaschi, S. Stuijk, T. Basten, and H. Corporaal. Online multi-face detection and tracking using detector confidence and structured SVMs. In *IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS)*, 2015. 3

[14] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):681–685, 2001. 3

[15] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995. 3

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

[17] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *International Conference on Machine learning*, pages 209–216, 2007. 3

[18] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013. 1, 2

[19] M. Du and R. Chellappa. Face association across unconstrained video frames using conditional random fields. In *European Conference on Computer Vision (ECCV)*, 2012. 3

[20] S. Duffner and J. Odobez. Track creation and deletion framework for long-term online multiface tracking. *IEEE Transactions on Image Processing*, 22(1):272–285, Jan. 2013. 2

[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014. 1

[22] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2014. 2

[23] P. Grother and M. Ngan. Face recognition vendor test(frvt): Performance of face identification algorithms. *NIST Interagency Report 8009*, 2014. 7

[24] 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. 5

[25] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 5, 6, 7

[26] 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. 1, 3, 4

[27] R. Ranjan, V. M. Patel, and R. Chellappa. A deep pyramid deformable part model for face detection. In *IEEE International Conference on Biometrics: Theory, Applications and Systems*, 2015. 1, 2, 3

[28] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1685–1692, June 2014. 4

[29] G. Ross. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015. 2

[30] M. Roth, M. Bauml, R. Nevatia, and R. Stiefelhagen. Robust multi-pose face tracking by multi-stage tracklet association. In *International Conference on Pattern Recognition (ICPR)*, 2012. 2

[31] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *arXiv preprint arXiv:1503.03832*, 2015. 1, 3

[32] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994. 4

[33] 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. 1, 7

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[35] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3476–3483, 2013. 3

[36] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. *arXiv preprint arXiv:1412.1265*, 2014. 3

[37] 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. 1, 5

[38] Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *British Machine Vision Conference*, pages 1–12, 2009. 3

[39] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014. 3

[40] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. 2, 6

[41] D. Wang, C. Otto, and A. K. Jain. Face search at scale: 80 million gallery. *arXiv preprint arXiv:1507.07242*, 2015. 7

[42] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 1, 5

[43] J. H. Yoon, M.-H. Yang, J. Lim, and K.-J. Yoon. Bayesian multi-object tracking using motion context from multiple objects. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015. 2

[44] W. Y. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003. 1

[45] X. G. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2879–2886. IEEE, 2012. 2, 3