This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2700390, IEEE Transactions on Pattern Analysis and Machine Intelligence

Trunk-Branch Ensemble Convolutional Neural Networks for Video-based Face Recognition

Changxing Ding, *Student Member, IEEE*, Dacheng Tao, *Fellow, IEEE*

Abstract—Human faces in surveillance videos often suffer from severe image blur, dramatic pose variations, and occlusion. In this paper, we propose a comprehensive framework based on Convolutional Neural Networks (CNN) to overcome challenges in video-based face recognition (VFR). First, to learn blur-robust face representations, we artificially blur training data composed of clear still images to account for a shortfall in real-world video training data. Using training data composed of both still images and artificially blurred data, CNN is encouraged to learn blur-insensitive features automatically. Second, to enhance robustness of CNN features to pose variations and occlusion, we propose a Trunk-Branch Ensemble CNN model (TBE-CNN), which extracts complementary information from holistic face images and patches cropped around facial components. TBE-CNN is an end-to-end model that extracts features efficiently by sharing the low- and middle-level convolutional layers between the trunk and branch networks. Third, to further promote the discriminative power of the representations learnt by TBE-CNN, we propose an improved triplet loss function. Systematic experiments justify the effectiveness of the proposed techniques. Most impressively, TBE-CNN achieves state-of-the-art performance on three popular video face databases: PaSC, COX Face, and YouTube Faces. With the proposed techniques, we also obtain the first place in the BTAS 2016 Video Person Recognition Evaluation.

Index Terms—Video-based face recognition, video surveillance, blur- and pose-robust representations, convolutional neural networks.

1 Introduction

With the widespread use of video cameras for surveillance and mobile devices, an enormous quantity of video is constantly being captured. Compared to still face images, videos usually contain more information, e.g., temporal and multi-view information. The ubiquity of videos offers society far-reaching benefits in terms of security and law enforcement. It is highly desirable to build surveillance systems coupled with face recognition techniques to automatically identify subjects of interest. Unfortunately, the majority of existing face recognition literature focuses on matching in still images, and video-based face recognition (VFR) research is still in its infancy [1], [2]. In this paper, we handle the still-to-video (S2V), video-to-still (V2S), and video-to-video (V2V) matching problems, which are used in the most common VFR applications.

Compared to still image-based face recognition (SIFR), VFR is significantly more challenging. Images in standard SIFR datasets are usually captured under good conditions or even framed by professional photographers, e.g., in the Labeled Faces in the Wild (LFW) database [3]. In comparison, the image quality of video frames tends to be significantly lower and faces exhibit much richer variations (Fig. 1) because video acquisition is much less constrained. In particular, subjects in videos are usually mobile, resulting in serious motion blur, out-of-focus blur, and a large range of pose variations. Furthermore, surveillance and mobile cameras are often low-cost (and therefore low-quality) devices, which further exacerbates problems with video frame clarity [4].

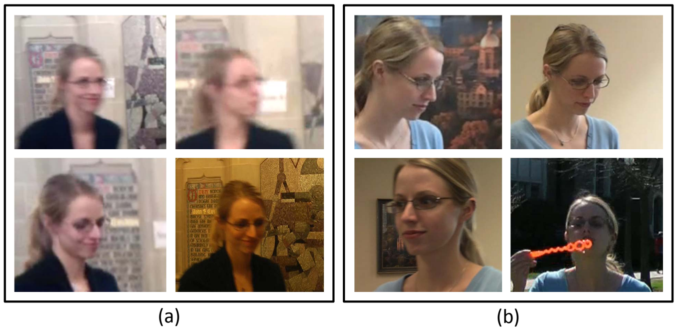


Fig. 1. Video frames captured by surveillance or mobile devices suffer from severe image blur, dramatic pose variations, and occlusion. (a) Image blur caused by the motion of the subject, camera shake (for mobile devices), and out-of-focus capture. (b) Faces in videos usually exhibit occlusion and a large range of pose variations.

Recent advances in face recognition have tended to ignore the peculiarities of videos when extending techniques from SIFR to VFR [5], [6], [7], [8]. On the one hand, a major difficulty in VFR, such as severe image blur, is largely unsolved [9]. One important reason is that large amounts of real-world video training data are still lacking, and existing still image databases are usually blur-free. On the other hand, although pose variations and occlusion are partially solved in SIFR by ensemble modelling [6], [10], the strategy may not be directly ex-

C. Ding and D. Tao are with the Centre for Artificial Intelligence, and the Faculty of Engineering and Information Technology, University of Technology Sydney, 81 Broadway, Ultimo, NSW 2007, Australia (email: changxing.ding@student.uts.edu.au, dacheng.tao@uts.edu.au).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2700390, IEEE Transactions on Pattern Analysis and Machine Intelligence

tended to VFR. A common practice in model ensembles is to train models separately for the holistic face image and for patches cropped around facial components. Model fusion is then performed offline at the feature or score level [11]. However, the accuracy promoted by model ensembles is at the cost of significantly increased time cost, which is impractical for VFR since each video usually contains dozens or even thousands of frames.

Here we approach the blur-robust representation learning problem from the perspective of training data. Since the volume of real-world video training data is small, we propose simulating large amounts of video frames from existing still face image databases. During training, we provide CNN with two training data streams: one composed of still face images, and the other composed of simulated video frames created by applying random artificial blur to the first stream. The network aims to classify each still image and its artificially blurred version into the same class; therefore, the learnt face representations must be blur-insensitive. To the best of our knowledge, this is the first CNN-based approach to solve the image blur problem in VFR.

To learn pose- and occlusion-robust representations for VFR efficiently, we propose a novel end-to-end ensemble CNN model called Trunk-Branch Ensemble CNN (TBE-CNN). TBE-CNN includes one trunk network and several branch networks. The trunk network learns face representations for holistic face images, and each branch network learns representations for image patches cropped around one facial component. To speed up computation, the trunk and branch networks share the same low- and middle-level layers, while their high-level layers are optimized independently. This sharing strategy significantly reduces the computational cost of the ensemble model and at the same time exploits each model's uniqueness. The output feature maps by the trunk network and branch networks are fused by concatenation to form a comprehensive face representation.

Furthermore, to enhance TBE-CNN's discriminative power, we propose a Mean Distance Regularized Triplet Loss (MDR-TL) function to train TBE-CNN in an end-to-end fashion. Compared with the popular triplet loss function [5], MDR-TL takes full advantage of label information and regularizes the triplet loss so that inter- and intra-class distance distributions become uniform.

The efficacy of the proposed algorithm is systematically evaluated on three popular video face databases: PaSC [12], COX Face [1], and YouTube Faces [13]. The evaluation is conducted for S2V, V2S, and V2V tasks. Extensive experiments on the three datasets indicate that the proposed algorithm achieves superior performance.

The remainder of the paper is organized as follows. Section 2 briefly reviews related VFR works. The TBE-CNN model that handles image blur and pose variations for VFR is described in Section 3. The TBE-CNN training strategy is introduced in Section 4. Face matching using the proposed approach is illustrated in Section 5. Results of experiments are presented in Section 6, leading to conclusions in Section 7.

2 Related Works

We review the literature in two parts: 1) video-based face recognition, and 2) deep learning methods for face recognition.

2.1 Video-based Face Recognition

Existing studies on VFR can be divided into three categories: (i) approaches for frame quality evaluation, (ii) approaches that exploit redundant information contained between video frames, and (iii) approaches that attain robust feature extraction from each frame.

Frame quality evaluation is mainly utilized for key frame selection from video clips, such that only a subset of best-quality frames is selected for efficient face recognition. For a systematic summary of frame quality evaluation methods, we direct readers to two recent works [14], [15].

In contrast to frame quality evaluation methods, a number of recent VFR studies attempt to make use of redundant information contained between video frames. Algorithms that fall in this category include sequence-based methods, dictionary-based methods, and image set-based methods [16]. The sequence-based methods aim to extract person-specific facial dynamics from continuous video frames [17], [18], which means that they rely on robust face trackers. The dictionary-based methods construct redundant dictionaries using video frames and employ sparse representation-based classifiers for classification [19], [20]. Due to the large size of the constructed dictionaries, the dictionary-based methods are often inefficient. The image set-based methods model the distribution of video frames using various techniques, e.g., affine/convex hull [21], [22], linear subspace [1], and manifold methods [23], [24]. They then measure the between-distribution similarity to match two image sets. The downside of image set modeling is that it is sensitive to the variable volume of video frames and complex facial variations that exist in real-world scenarios [25].

Extracting high-quality face representations has always been a core task in face recognition [26], [27]. In contrast to still face images, video frames usually suffer from severe image blur because of the relative motion between the subjects and the cameras. Two types of methods have been introduced to reduce the impact of image blur: deblur-based methods [28] and blur-robust feature extraction-based methods [29], [30]. The former method first estimates a blur kernel from the blurred image and then deblurs the face image prior to feature extraction. However, the estimation of the blur kernel is challenging, as it is an ill-posed problem. Of the blur-robust feature extraction methods, Ahonen *et al.* [31] proposed to employ the blur-insensitive Local Phase Quantization (LPQ) descriptor for facial feature extraction, which has been widely used in VFR applications [9]. However, to the best of our knowledge, no

CNN-based method has yet been used to handle the image blur problem in VFR. Furthermore, similar to still face images, faces in video frames exhibit rich pose, illumination, and expression variations and occlusion; therefore, existing studies tend to directly extend feature extractors designed for SIFR to VFR [32], [33], [34].

Our proposed approach falls into the third category of methods. Compared to previous VFR approaches, we propose an efficient CNN model to automatically learn face representations that are robust to image blur, pose variations, and occlusion.

2.2 Deep Learning Methods for Face Recognition

Representation learning with CNNs provides an effective tool for face recognition. Promising results have been obtained, but they are usually limited to SIFR [5], [7], [35]. For example, Taigman *et al.* [36] formulated face representation learning as a face identification problem in CNN. [11], [37], [5], [38] proposed deep metric learning methods to enhance the discriminative power of learnt face representations. In comparison, there are only limited studies on CNN-based VFR.

This is for a number of reasons. First, existing CNN-based approaches do not handle the peculiarities of video frames (such as severe image blur) very well. Training from real-world video data is an intuitive solution for VFR. However, existing face video databases are rather small, e.g., 2,742 videos in PaSC and 3,000 videos in COX Face. Second, the video data are highly redundant, because frames in the same video tend to be similar in image quality, expression, occlusion, and illumination conditions. Therefore, direct CNN training using real-world video data is prone to overfitting. As a result, the majority of existing works employ CNN models trained on large-scale still image databases for VFR [36], [7] and ignore the difference in image quality between still images and video frames. Recently, Huang *et al.* [9] proposed pre-training CNN models with a large volume of still face images and then fine-tuning the CNN models with small real-world video databases. However, the fine-tuning strategy is suboptimal, as it only slightly adapts CNN parameters to the video data. In comparison, our proposed video data simulation strategy enables direct optimization of all model parameters for VFR. Video data simulation can be regarded as a novel method for data augmentation. Compared to previous augmentation methods such as horizontal flipping and translation, video data simulation produces visually different images compared to the original still images, as illustrated in Fig. 2.

Part-based models have been widely used in both detection [39], [40], [41], [42], [43] and recognition [44], [45], [46] tasks to account for occlusion, pose variations, and non-rigid object deformations. For SIFR [11], [10], the principle is to train CNN models separately using the holistic face image and patches cropped around facial components. Model fusion is usually conducted offline by directly concatenating representations learnt by all models. Since image patches are less sensitive to pose variations and occlusion [47], the ensemble system outperforms single models. However, there are drawbacks from two perspectives. First, the ensemble system is not an end-to-end system, which means its parameters are not simultaneously optimized for the VFR task. Second, extracting features separately from multiple models significantly reduces efficiency, which may be impractical for VFR since each video tends to have a number of frames to process.

A few recent works for fine-grained image classification are relevant to this paper [44], [45], [46]. Compared with [46], TBE-CNN is more concise and more efficient: it integrates the networks for the holistic image and all facial components into a single model by sharing their low- and middle-level layers. Besides, [44], [45], [46] dictate that object parts share all convolutional layers and a representation of each part is formed by cropping feature maps from the last convolutional layer of CNN. However, feature maps extracted by the high-level convolutional layers represent abstract and global information; therefore, it may be less effective to utilize it to compute the representations of facial components. In comparison, TBE-CNN is more discriminative: the holistic image and each facial component have independent high-level convolutional layers to better exploit their complementary information.

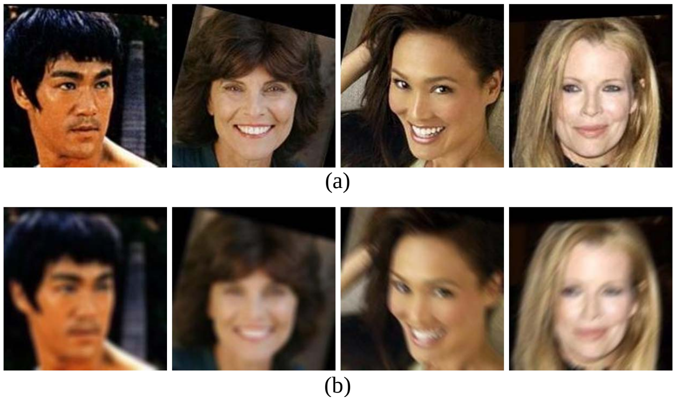
3 Trunk-Branch Ensemble CNN for VFR

In this section, we make two contributions to VFR. First, we introduce a method for artificially simulating video-like training image data for blur-robust face recognition. Second, the efficient TBE-CNN model for VFR is described. As we lack real-world video sequences, the training set of TBE-CNN is composed of a large amount of face images. For testing in VFR, TBE-CNN is employed to extract features from individual video frames and fuse their features by average pooling, detailed in Section 5.

3.1 Artificially Simulated Video Data

As described above, most available video face databases are rather small and lack diversity in facial variations compared to still face image databases. We propose artificially generating video-like face data from existing large-scale still face image databases. Specifically, we simulate two major challenges during surveillance or mobile camera imaging: motion blur and out-of-focus blur. For simplicity, we assume the blur kernels are uniform since faces usually occupy a small area in video frames.

Due to face movement or mobile device camera shake during exposure, motion blur often appears in video frames. Supposing the relative motion is along a single


(a)

(b)

Fig. 2. Examples of the original still face images and simulated video frames. (a) original still images; (b) simulated video frames by applying artificial out-of-focus blur (the two figures on the left) and motion blur (the two figures on the right).

direction during exposure, we can model the motion blur effect by one-dimensional local averaging of neighboring pixels using the following kernel:

$$k_m(i,j;L,\theta) = \begin{cases} \frac{1}{L}, & \text{if } \sqrt{i^2+j^2} \leq \frac{L}{2} \text{ and } \frac{i}{j} = -\tan\theta, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where (i,j) is the pixel coordinate originating from the central pixel, L is the size of the kernel and indicates the motion distance during exposure, and θ denotes the motion direction [48].

Due to the limited depth of field (DOF) of cameras and the large motion range of faces in videos, out-of-focus blur also occurs in video frames and can be simulated using a uniform kernel [48] or a Gaussian kernel [28]. In this paper, we employ the Gaussian kernel in the following form:

$$k_o(i,j) = \begin{cases} C \cdot \exp\left(-\frac{i^2+j^2}{2\sigma^2}\right), & \text{if } i \leq \frac{R}{2} \text{ and } j \leq \frac{R}{2}, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where σ and R denote the magnitude and size of the kernel, respectively. C is a constant which ensures the kernel has a unit volume.

Given one still face image I_s and one blur kernel k, the simulated video frame I_v can be obtained by convolution:

$$I_v = I_s * k. \tag{3}$$

Based on the above description, we blur each still face image using one randomly sampled blur type from the following 38 possibilities. Specifically, we choose the value of L from $\{7, 9, 11\}$ and the value of θ from $\{0, \pi/4, \pi/2, 3\pi/4\}$; therefore, there are 12 motion blur choices. Similarly, we set the value of R as 9 and randomly choose the value of σ from $\{1.5, 3.0\}$; i.e., there are 2 choices for out-of-focus blur. We also enrich the blur types by sequentially conducting out-of-focus blur and motion blur, of which there are 24 combinations. Samples of original still images and the corresponding simulated video frames are illustrated in Fig. 2. Since we obtain one blurred image from each still image, we obtain two training data streams of equal size, i.e., one stream composed of the original still images and the other stream composed of the same number of blurred images. For CNN training, we provide both training data streams to CNN simultaneously. Since we encourage each still image and its blurred version to be classified into the same class, CNN automatically learns blur-robust face representations.

It is worth noting that more realistic blur kernels can be found in the literature of deblur. For example, Sun *et al.* [49] estimated the non-uniform motion blur kernel field from a large field-of-view image. But it is not clear how to artificially blur images using non-uniform motion blur kernel fields. Xu *et al.* [50] presented a complicated blur kernel that considers saturation, camera noise, and compression artifacts. We empirically found that the performance gain obtained by the blur kernel in [50] is slight on the PaSC database compared with the blur kernels in Eq. 1 and Eq. 2. Therefore, we keep using the blur kernels in Eq. 1 and Eq. 2 in this paper.

3.2 Trunk-Branch Ensemble CNN

We propose the Trunk-Branch Ensemble CNN (TBE-CNN) model to efficiently learn pose- and occlusion-robust face representations. TBE-CNN incorporates one trunk network and several branch networks. The trunk network is trained to learn face representations for holistic face images, and each branch network is trained to learn face representations for image patches cropped from one facial component. In this paper, the trunk network implementation is based on GoogLeNet [51] and results of experiments based on VGG Net [52] are available in the supplementary appendix. The most important parameters of the trunk network are tabulated in Table 1. For the other model parameters, we directly follow its original configuration [51]. As shown in Table 1, we divide the GoogLeNet layers into three levels: the low-level layers, middle-level layers, and high-level layers. The three layer levels successively extract features from the low- to the high-level. Since low- and middle-level features represent local information, the trunk network and branch networks can share low- and middle-level layers. In comparison, high-level features represent abstract and global information; therefore, different models should have independent high-level layers. Based on the above observations, the TBE-CNN architecture is illustrated in Fig. 3.

The trunk network extracts features from raw pixels. On top of Inception 5 of the trunk network, we reduce the size of its feature maps to 3×3×1024 by max pooling. For each branch network, we directly crop feature maps from Conv2 and Inception 3 module outputs of the trunk network instead of computing low- and middle-level features from scratch. The cropping size and position

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2700390, IEEE Transactions on Pattern Analysis and Machine Intelligence

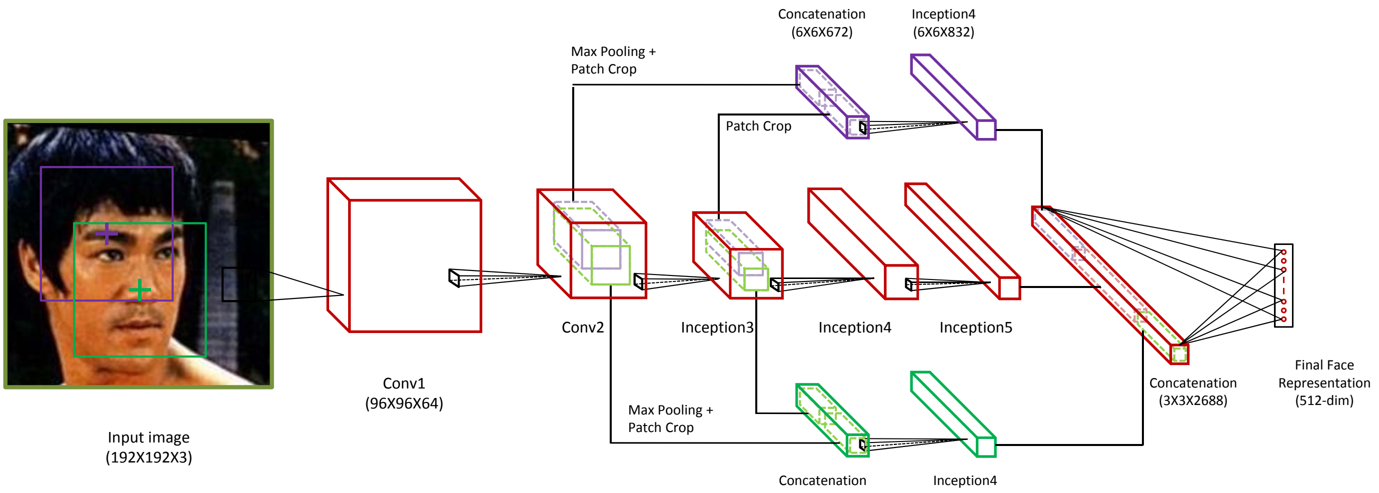


Fig. 3. Model architecture for Trunk-Branch Ensemble CNN (TBE-CNN). Note that a max pooling layer is omitted for simplicity following each convolution module, e.g., Conv1 and Inception 3. TBE-CNN is composed of one trunk network that learns representations for holistic face images and two branch networks that learn representations for image patches cropped around facial components. The trunk network and the branch networks share the same low- and middle-level layers, and they have individual high-level layers. The output feature maps of the trunk network and branch networks are fused by concatenation. The output of the last fully connected layer is utilized as the final face representation of one video frame.

TABLE 1
Trunk Network Parameters (GoogLeNet)

	Type (Name)	Kernel Size/ Stride	Output Size	Depth
Low-Level	convolution (Conv1)	7×7/2	96×96×64	1
	max pool	2×2/2	48×48×64	0
	convolution (Conv2)	3×3/1	48×48×192	2
	max pool	2×2/2	24×24×192	0
Middle-L	inception (3a)	-	24×24×256	2
	inception (3b)	-	24×24×480	2
	max pool	2×2/2	12×12×480	0
High-Level	inception (4a)	-	12×12×512	2
	inception (4b)	-	12×12×512	2
	inception (4c)	-	12×12×512	2
	inception (4d)	-	12×12×528	2
	inception (4e)	-	12×12×832	2
	max pool	2×2/2	6×6×832	0
	inception (5a)	-	6×6×832	2
	inception (5b)	-	6×6×1024	2
	max pool	2×2/2	3×3×1024	1
	Dropout (0.4)	-	3×3×1024	1
	Fully-connected	-	512	1

are propositional to those of the patches of interest from the input image, as illustrated in Fig. 3. The size of the feature maps cropped from the Conv2 output is reduced by a half by max pooling and then concatenated with the feature maps cropped from the Inception 3 output. The concatenated feature maps form the input of the branch network. To promote efficiency, each branch network includes only one Inception 4 module. Similar to the trunk network, the size of feature maps of the branch network is reduced to 3×3×832 by max pooling.

We include two branch networks in TBE-CNN, as illustrated in Fig. 3. The output feature maps of the trunk network and branch networks are fused by concatenation to form an over-complete face representation, whose dimension is reduced to 512 by one fully connected layer. The 512-dimensional feature vector is utilized as the final face representation of one video frame.

4 TBE-CNN Training

We propose a stage-wise training strategy to effectively optimize the TBE-CNN model parameters. In the first stage, the trunk network illustrated in Table 1 is trained alone using softmax loss. In the second stage, the trunk network parameters are fixed, and each of the branch networks is trained with softmax loss[1]. After the trunk network and all branch networks are pre-trained, we fine-tune the complete model illustrated in Fig. 3 with softmax loss to fuse the trunk and branch networks. Finally, to enhance the discriminative power of learnt face representations, we propose a novel deep metric learning method called Mean Distance Regularized Triplet Loss (MDR-TL) to fine-tune the complete network.

The softmax loss has the advantage of convergence efficiency, but the penalty is imposed on the classification accuracy on training data rather than the discriminative power of learnt face representations. In comparison, the convergence rate of MDR-TL or the triplet loss is slower, but they directly optimize the discriminative power of the face representations. In the following, we first briefly

[1] We include one 256-dimensional fully connected layer after the Inception 4 module of each branch network when training it alone.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2700390, IEEE Transactions on Pattern Analysis and Machine Intelligence

review the triplet loss for CNN training [53], [5]. Then, we introduce the principle of MDR-TL. For detailed experimental setting about MDR-TL and the triplet loss, e.g., triplet sampling method and batch size, one can refer to Section 6.3.

4.1 Brief Review of the Triplet Loss

Schroff *et al.* [5] proposed to employ the triplet loss to train CNNs for face recognition. In [5], the representation of a face image is ℓ_2-normalized as the input of the triplet loss function; therefore, the input face representations for the triplet loss lie on a unit hypersphere. We denote the ℓ_2-normalized face representation of one image x as $f(x) \in \mathbb{R}^d$. For accurate face recognition, it is desired that the representation of an image $f(x^a)$ (anchor) of a specific subject is closer to each image representation $f(x^p)$ (positive) of the same subject than it is to any image representation $f(x^n)$ (negative) of other subjects. $\{f(x^a), f(x^p), f(x^n)\}$ compose a triplet. Formally, each triplet is desired to satisfy the triplet constraint [5]:

$$\|f(x^a) - f(x^p)\|_2^2 + \beta < \|f(x^a) - f(x^n)\|_2^2, \tag{4}$$

where β is the margin that is enforced between the positive pair $\{f(x^a), f(x^p)\}$ and the negative pair $\{f(x^a), f(x^n)\}$. The triplet loss function is formulated as $L_{triplet}(f) =$

$$\frac{1}{2N}\sum_{i=1}^{N}\left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \beta\right]_+ , \tag{5}$$

where N is the number of triplets that violate the triplet constraint in a batch and $\{f(x_i^a), f(x_i^p), f(x_i^n)\}$ stands for the i-th triplet. In this paper, triplets are sampled online within each batch.

4.2 Mean Distance Regularized Triplet Loss

Both the pairwise loss [11] and the triplet loss [53], [5] rely on sampling effective image pairs or triplets in a batch from possible combinations. Take the triplet loss for example, the optimization of $L_{triplet}(f)$ is based on each individual triplet $\{f(x_i^a), f(x_i^p), f(x_i^n)\}(1 \leq i \leq N)$ in Eq. 5. However, the inter- and intra-class distance distributions are not explicitly constrained, which has a negative impact on face recognition. For example, the training samples in Fig. 4(a) satisfy the triplet constraint in Eq. 4; but due to the non-uniform inter- and intra-class distance distributions, it is difficult to find an ideal threshold in face verification to determine whether a pair of face images belong to the same subject or not. To overcome this problem, we propose the MDR-TL loss function, which regularizes the triplet loss by constraining the distance between subject mean representations.

Similar to [5], the 512-dimensional face representation extracted by TBE-CNN is ℓ_2-normalized as the MDR-TL input, as illustrated in Fig. 5. Two constraints are included in MDR-TL. In the first, the mean representations of different subjects should be well separated to ensure that the inter-class distance distribution is uniform. We realize this constraint by enforcing a margin α between the mean representation $\hat{\mu}_c$ of one subject and its nearest mean representation $\hat{\mu}_c^n$ in the same batch:

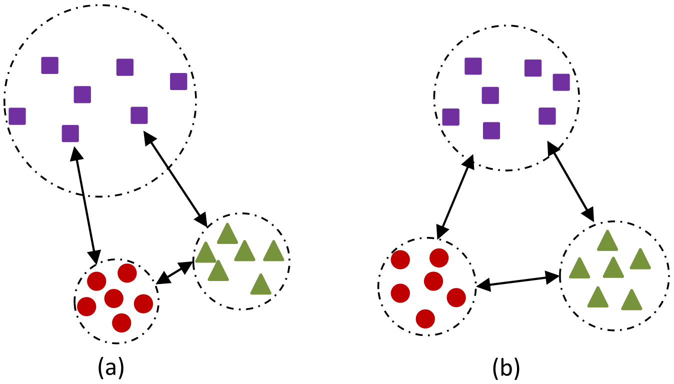


Fig. 4. The principle of Mean Distance Regularized Triplet Loss (MDR-TL). (a) Triplets sampled in the training batch satisfy the triplet constraint (Eq. 4). However, due to the non-uniform intra- and inter-class distance distributions, it is hard to select an ideal threshold for face verification. (b) MDR-TL regularizes the triplet loss by setting a margin for the distance between subject mean representations.

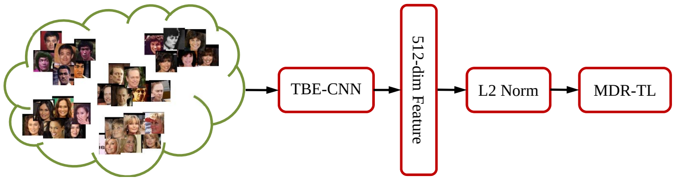


Fig. 5. Illustration of TBE-CNN training with MDR-TL. MDR-TL is employed to further enhance the discriminative power of learnt face representations.

$$\|\hat{\mu}_c - \hat{\mu}_c^n\|_2^2 > \alpha, \tag{6}$$

where $\hat{\mu}_c$ is the Karcher mean (Riemannian geometric mean) [54] that also lies on the unit hypersphere. For simplicity, we compute $\hat{\mu}_c$ in the following way:

$$\hat{\mu}_c = \frac{\mu_c}{\|\mu_c\|}, \tag{7}$$

and

$$\mu_c = \frac{1}{N_c}\sum_{i=1}^{N_c} f(x_{ci}), \tag{8}$$

where N_c is the number of images of the c-th subject in the current batch and $f(x_{ci})$ denotes the face representation of the i-th image for the c-th subject. The nearest mean representation $\hat{\mu}_c^n$ for $\hat{\mu}_c$ is detected online efficiently within the same batch.

In the second, triplets sampled in a training batch should satisfy the triplet constraint in Eq. 4. Working together with the first constraint, the triplet constraint is helpful to regularize the intra-class distance distribution to be uniform. Based on the above analysis, we formulate MDR-TL as the following optimization problem:

$$\min_{f} L(f) = L_{triplet}(f) + L_{mean}(f), \tag{9}$$

where

$$L_{mean}(f) = \frac{1}{2P}\sum_{c=1}^{C}\max(0, \alpha - \|\hat{\mu}_c - \hat{\mu}_c^n\|_2^2). \tag{10}$$

In Eq. 9, we assume equal weights for $L_{triplet}(f)$ and $L_{mean}(f)$, which empirically works well. In Eq. 10, C is the number of subjects in the current batch, and P is the number of mean representations that violate the constraint in Eq. 6. We optimize Eq. 9 using the standard stochastic gradient descent with momentum [55]. The gradient of L with respect to $f(x_{ci})$ is derived as follows,

$$\frac{\partial L}{\partial f(x_{ci})} = \frac{\partial L_{triplet}}{\partial f(x_{ci})} + \frac{\partial L_{mean}}{\partial f(x_{ci})}, \tag{11}$$

where

$$\frac{\partial L_{mean}}{\partial f(x_{ci})} = -\frac{1}{P}\Big(\sum_{j=1,j\neq c}^{C} w_j(\hat{\mu}_c - \hat{\mu}_j)^T\Big)\frac{\partial \hat{\mu}_c}{\partial f(x_{ci})}. \tag{12}$$

w_j equals 1 if the constraint in Eq. 6 is violated, and $\hat{\mu}_j$ is the nearest neighbor of $\hat{\mu}_c$ or $\hat{\mu}_c$ is the nearest neighbor of $\hat{\mu}_j$. Otherwise, w_j equals 0.

$$\begin{aligned}\frac{\partial \hat{\mu}_c}{\partial f(x_{ci})} &= \|\mu_c\|^{-1}\frac{\partial \mu_c}{\partial f(x_{ci})} + \mu_c\frac{\partial \|\mu_c\|^{-1}}{\partial f(x_{ci})}\\ &= \frac{1}{N_c\|\mu_c\|}(I - \hat{\mu}_c\hat{\mu}_c^T).\end{aligned} \tag{13}$$

The derivative of L with respect to model parameters can be computed via the chain rule. As illustrated in Fig. 4(b), by setting a margin between the mean representations of different subjects, the inter- and intra-class distance distributions become uniform. This regularization is empirically verified as important below.

5 VFR WITH TBE-CNN

We next employ the TBE-CNN model for VFR. Given a video, we first augment its video frames by horizontal flipping. Then, all video frames pass through the TBE-CNN network, and fusing their outputs by average pooling produces the compact video representation. The representation of one still image can be extracted in a similar way by assuming it is a single-frame video. For all three settings - S2V, V2S, and V2V matching - we consistently employ the cosine metric to calculate the similarity ρ between the two representations y_1 and y_2:

$$\rho\,(y_1, y_2) = \frac{y_1^T y_2}{\|y_1\|\|y_2\|}. \tag{14}$$

6 EXPERIMENTS

We now systematically evaluate the proposed TBE-CNN framework for VFR. Experiments are conducted on three publicly available large-scale video face databases: PaSC, COX Face, and YouTube Faces. Example images are shown in Fig. 6.

The PaSC database [12] contains 2,802 videos of 265 subjects. The videos were recorded by multiple sensors in varied indoor and outdoor locations. They are divided into two sets: a control set and a handheld set. High-end cameras installed on tripods captured videos in the control set to produce images of good quality, as illustrated by the first four images in the first row in Fig. 6. Videos in the handheld set were captured by five handheld video cameras. For each video, the subject was asked to carry out actions to create a wide range of poses at variable distances from the camera. Faces in the video exhibit serious motion and out-of-focus blur and rich pose variations. Here, we adopted the officially defined V2V matching protocol for face verification [12].

The COX Face database [1] incorporates 1,000 still images and 3,000 videos of 1,000 subjects. A high-quality camera in well-controlled conditions captured still images to simulate ID photos. The videos were taken while the subjects were walking in a large gym to simulate surveillance. Three cameras at different locations were installed to capture videos of the walking subject simultaneously. Videos captured by the three cameras created three subsets: Cam1, Cam2, and Cam3. Since COX Face has significantly more subjects compared to PaSC, it is an ideal database for face identification. The standard S2V, V2S, and V2V matching protocols [1] for face identification were adopted.

The YouTube Faces database [13] includes 3,425 videos of 1,595 subjects. All videos were downloaded from the YouTube website. Since the majority of subjects in this database were in interviews, there is no obvious image blur or pose variation. Instead, this database is low resolution and contains serious compression artifacts, as shown in the last four video frames of the third row in Fig. 6. 5,000 video pairs were collected from the videos, which were divided into 10 equally sized splits. Each split contained 250 homogeneous video pairs and 250 heterogeneous video pairs. Following the “restricted” protocol defined in [13], we report the mean verification accuracy and the standard error of the mean (S_E) on the 10 splits.

For all three video databases, we employ the face detection results provided by the respective databases. There are 60 videos in PaSC that have no face being detected. We ignore these 60 videos since their similarity score to any video is NaN, and thus have no impact on the verification rate or Receiver Operating Characteristic (ROC) curve. Each detected face is normalized and resized to 192×192 pixels using an affine transformation based on the five facial feature points detected by [56]. Sample face images after normalization are shown in

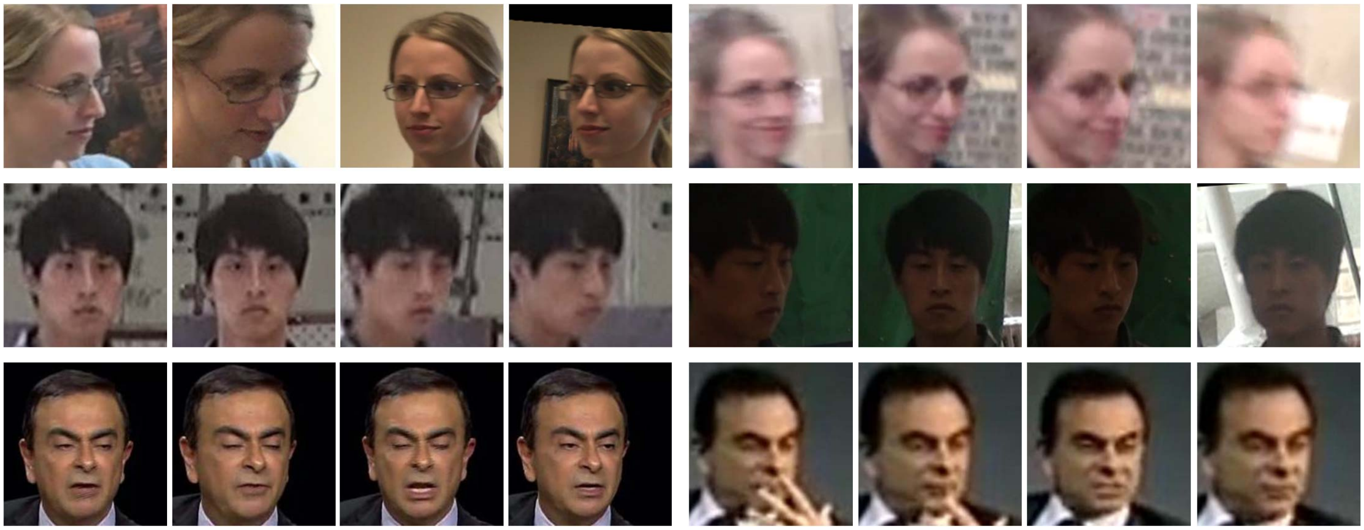


Fig. 6. Sample video frames after normalization: PaSC (first row), COX Face (second row), and YouTube Faces (third row). For each database, the four frames on the left are sampled from a video recorded under relatively good conditions, and the four frames on the right are selected from low-quality video.

Fig. 6. A number of experiments are conducted. First, the effectiveness of video training data simulation for CNN is tested. Second, the MDR-TL performance is evaluated. Third, the trunk network performance and TBE-CNN are compared. Finally, the performance of the complete TBE-CNN framework is compared to state-of-the-art VFR methods on the PaSC, COX Face, and YouTube Faces databases.

6.1 Implementation Details of TBE-CNN

The proposed TBE-CNN model is trained on the publicly available CASIA-WebFace database [57] and deployed in experiments on the three video face databases described above. The CASIA-WebFace database contains 494,414 images of 10,575 subjects. We augment the CASIA-WebFace database by horizontal flipping and image jittering[2]. After augmentation, the size of the training set is about 2.68 million, which forms the still image training data stream. According to the description in Section 3, we blur each still image using a randomly chosen kernel to form the simulated video training data stream.

During training, the batch size is set to 90 for both the still image stream and simulated video stream to produce a complete batch size of 180. Following the stage-wise training framework introduced in Section 4, the trunk network is trained for 13 epochs using softmax loss, with the learning rate gradually decreased from 0.01 to 0.001. Then, the branch networks are trained for 4 epochs. Next, the complete TBE-CNN model is fine-tuned for another 4 epochs with softmax loss to fuse the trunk and branch networks with a small learning rate of 0.001. Finally, the TBE-CNN model is fine-tuned with MDR-TL for one more epoch with the learning rate of 0.001. The open-source deep learning package Caffe [55] is utilized to train the deep models.

The first three experiments are conducted on PaSC. For these three experiments, we manually remove the falsely detected faces from the face detection results provided by the database[3] to help us to accurately reflect the effectiveness of proposed algorithms. For the fourth experiment, we provide results based on both the original face detections and manually corrected face detections.

6.2 Effectiveness of Simulated Video Training Data

This experiment provides evidence to justify the proposed video data simulation strategy. The evaluation is based on the trunk network with softmax loss. Four types of training data are evaluated: the still image training data stream alone (SI), the simulated video training data stream alone (SV), the fine-tuning (FT) strategy adopted for VFR [9], and the two-stream training data (TS) that combines both SI and SV. For SI, SV, and TS, the networks are trained with the same number of iterations (13 epochs), and the amount of augmented training data is 2.6 million, 2.6 million, and 5.2 million, respectively. For FT, we fine-tune the network produced by SI with real-world video data. Following [9], we use all video frames in the COX Face database and the PaSC training set for fine-tuning. The two video sets incorporate 0.45 million video frames, comparable in volume to CASIA-WebFace. For fair comparison with TS, we augment the video frames with horizontal flipping and jittering to 2.6 million. Therefore, the total amount of training data for FT is 5.2 million. The learning rate of the fine-tuning

[2] For image jittering, we add random gaussian noise on the coordinates of the five facial feature points. The noise is distributed with zero mean and standard deviation of four pixels.

[3] The defect in face detection is because several different faces appeared in the same video. The list of manually removed faces is available upon request.

TABLE 2
Verification Rates (%) at 1% FAR on PaSC with Different Types of Training Data

	# Training Data (Augmented)	Control Set	Handheld Set
SI	2.6M	83.72	68.20
SV	2.6M	81.89	73.80
FT	5.2M	81.41	73.70
TS	5.2M	**87.31**	**79.33**

stage for FT is set to 0.001, and we observe that its performance saturates on PaSC after one epoch.

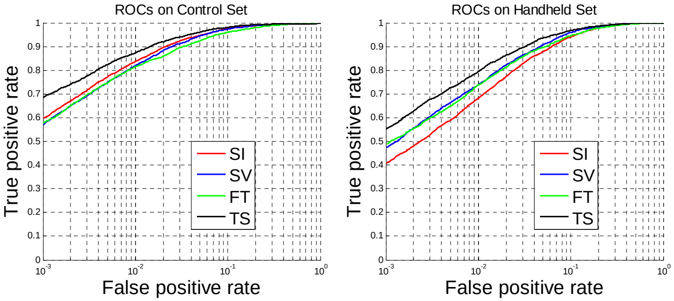
The verification rates at 1% False Acceptance Rate (FAR) of the above four types of training data are tabulated in Table 2, and the corresponding ROC curves are illustrated in Fig. 7. While SI slightly outperforms SV on the control set, SV significantly outperforms SI by as much as 5.6% on the handheld set. This is due to the difference in image quality between the control and handheld sets. Specifically, high-end cameras recorded video frames in the control set; therefore, the image quality is similar to that of the training data for SI. In comparison, low-quality mobile cameras captured videos in the handheld set and its frames sufferred from severe image blur, as illustrated in Fig. 6. The clear performance advantage of SV on the handheld set justifies the use of simulated video training data.

The performance of FT is comparable to SV, but at the cost of more training data and longer training time. In comparison, TS significantly outperforms SI, SV, and FT on both sets. Compared to SI and SV, TS combines the training data of both still images and simulated video frames, which regularizes the model to learn blur-robust representations. Compared to FT, which has the same size of training data, TS shows a performance advantage of 5.9% and 5.6% for the control and handheld sets, respectively. This is for two main reasons: first, there is only slight adaptation of CNN model parameters by FT to the VFR task compared to TS, which directly optimizes all model parameters of CNN for VFR; second, frames in the same video are highly redundant. Therefore, fine-tuning with a small amount of real-world video data may suffer from over-fitting.

6.3 Effectiveness of MDR-TL

This experiment evaluates the performance of the proposed MDR-TL loss function. Two baseline loss functions for deep metric learning are compared: the pairwise loss [58] and the triplet loss [5]. Similar to the previous experiment, performance comparisons are based on the trunk network. Two representative training data types are considered: 1) SI, which is the most widely used training data type; and 2) our proposed TS. The models produced in the previous experiment are fine-tuned with a learning rate of 0.001 for one epoch with each of the three loss functions. We observe that the performance saturates after one epoch of fine-tuning. Image pairs or triplets are sampled within each batch online. To ensure that sufficient image pairs or triplets are sampled, we re-arrange the training samples such that each subject in a batch contains 6 images. Therefore, there are 30 subjects in total for a batch size of 180. For the pairwise loss, our implementation is based on the Discriminative Locality Alignment (DLA) model [58], which has three parameters to tune. We empirically find that sampling all positive pairs and the same number of hard-negative pairs is optimal for DLA. For the triplet loss, following the specifications in [5], we utilize all positive image pairs and randomly sample semi-hard negative samples to compose triplets. We traverse the value of the margin parameter β within $\{0.1, 0.2, 0.3, 0.4\}$ and report its best performance. Similarly, we traverse the value of the margin parameter α for MDR-TL within $\{1.6, 1.8, 2.0, 2.2\}$ and report the best result. It is worth noting that our implementation of the $L_{triplet}(f)$ term in Eq. 9 for both the triplet loss and MDR-TL is exactly the same. The optimal values for α and β are 2.0 and 0.2, respectively. The verification rates at 1% FAR of the three loss functions are compared in Table 3.

It is shown that the performance of the pairwise loss (DLA) and the triplet loss are comparable with the SI training data. But the triplet loss visibly outperforms D-LA with the TS training data. One major disadvantage of DLA is that it has more parameters which substantially affect the performance. In comparison, the triplet loss has only one parameter and its optimal value is stable across experiments. MDR-TL outperforms the baselines



(a) (b)

Fig. 7. ROC curves of the trunk network trained with different types of training data on the PaSC database. (a) Control set; (b) handheld set.

TABLE 3
Verification rates (%) at 1% FAR with different loss functions on PaSC (Semi-hard Negative Mining)

	SI Training Data		TS Training Data	
	Control Set	Handheld Set	Control Set	Handheld Set
Pairwise loss	94.61	87.16	95.35	92.03
Triplet loss	94.54	87.00	96.03	92.46
MDR-TL	**95.86**	**89.60**	**96.66**	**93.91**

TABLE 4
Verification rates (%) at 1% FAR with different loss functions on PaSC (Hard Negative Mining)

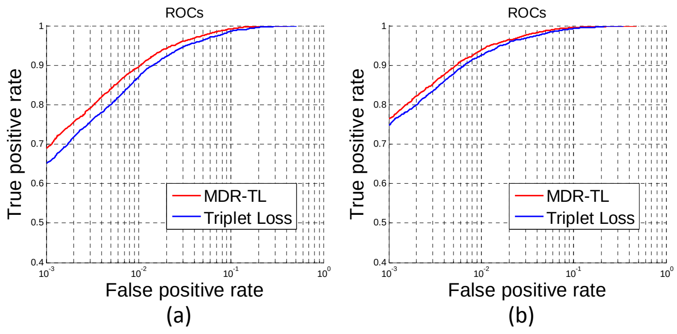
	SI Training Data		TS Training Data	
	Control Set	Handheld Set	Control Set	Handheld Set
Triplet loss	94.78	86.77	96.07	92.73
MDR-TL	**95.66**	**89.47**	**96.64**	**93.61**

TABLE 5
Verification rates (%) at 1% FAR with different loss functions on PaSC (Hardest Negative Mining)

	SI Training Data		TS Training Data	
	Control Set	Handheld Set	Control Set	Handheld Set
Triplet loss	93.86	85.35	95.22	91.47
MDR-TL	**95.32**	**88.35**	**96.00**	**93.12**

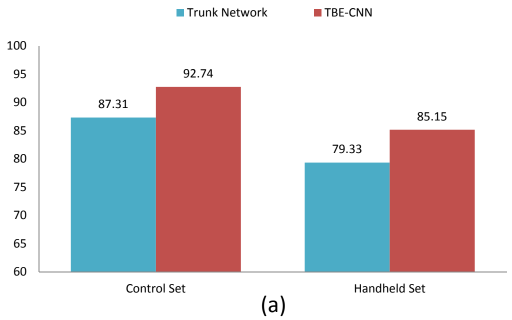
consistently under all settings. In particular, for the most challenging setting, i.e., the handheld set of PaSC with SI training data, MDR-TL outperforms the triplet loss by as much as 2.6%. With TS training data, the performance margin between MDR-TL and the triplet loss is around 1%, which means about 20% reduction in verification error rate. The ROC curves of the two loss functions on the handheld set of PaSC are plotted in Fig. 8. The results of experiments justify the role of $L_{mean}(f)$ in Eq. 9 as an effective regularization term to the triplet loss.

We also compared the performance of the triplet loss and MDR-TL with another two popular negative sampling strategies, i.e., hard negative sampling [5] and hardest negative sampling [59]. Results of experiments are tabulated in Table 4 and Table 5, respectively. It is clear that MDR-TL outperforms the triplet loss under all settings. It is also shown that the hardest negative sampling method is inferior to semi-hard negative mining and hard negative mining methods. Besides, the performance of hard negative mining and semi-hard negative mining is equivalent. This is because that the training set, i.e., CASIA-WebFace, was collected in a semi-automatic way; therefore, the labels of the training images are noisy. In this case, selecting the hardest negative sample is suboptimal due to the sensitivity to the labeling error.

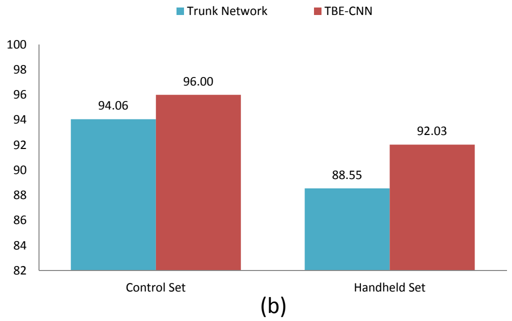


(a) (b)

Fig. 8. ROC curves of MDR-TL and the triplet loss functions on the handheld set of PaSC. (a) SI training data; (b) TS training data.

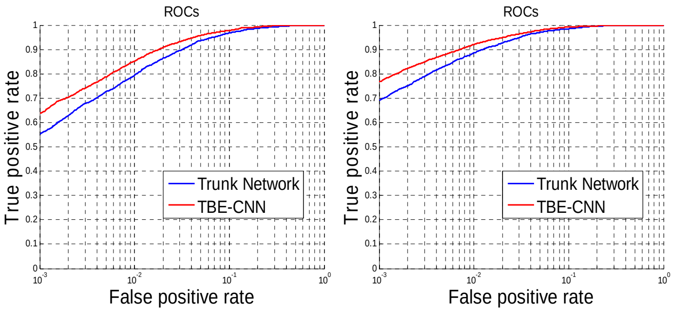


(a)



(b)

Fig. 9. Verification rates (%) at 1% FAR by the trunk network and TBE-CNN. Comparison is based on the softmax loss. (a) Performance comparison without BN layers; (b) performance comparison with BN layers.



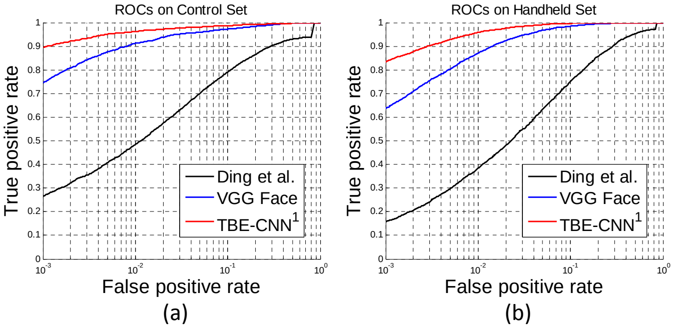
(a) (b)

Fig. 10. ROC curves of the trunk network and TBE-CNN on the handheld set of PaSC. (a) Without BN layers; (b) with BN layers.

The comparisons in Table 3, Table 4, Table 5, and Fig. 8 also convincingly show that the proposed TS strategy to compose CNN training data is essential to achieve blur-robustness in VFR. Briefly, for the control and handheld sets of PaSC, CNN performance with TS training data is higher than that with SI training data by around 1.0% and 5.0%, respectively.

6.4 Effectiveness of Trunk-Branch Fusion

The verification rates at 1% FAR of TBE-CNN and trunk network with TS training data on PaSC are compared



(a) (b)

Fig. 11. ROC curves of TBE-CNN and state-of-the-art methods on the PaSC control and handheld sets. The original face detection results from the database are employed for all methods. (a) Control set; (b) handheld set.

TABLE 6
Verification Rates (%) at 1% FAR of Different Methods on PaSC

	# Training Data (Original)	Control Set	Handheld Set
Ding *et al.* [9]	0.01M	48.00	38.00
Huang *et al.* [9]	0.5M	58.00	59.00
HERML [61]	0.5M	46.61	46.23
VGG Face [7]	2.62M	91.25	87.03
TBE-CNN^1	0.49M	**95.83**	**94.80**
TBE-CNN^1+BN	0.49M	**96.23**	**95.85**
TBE-CNN^2+BN	0.49M	**97.80**	**96.12**

in Fig. 9. Comparison is based on the softmax loss. Two results are presented, i.e., with and without the batch normalization (BN) [60] layers employed. To enable the training of TBE-CNN by a single GPU, we only add BN layers after convolutional layers of the Inception 4 and Inception 5 modules as BN consumes video memory. The performance of TBE-CNN is considerably higher than the trunk network. In particular, on the more challenging handheld set, the margin is as large as 5.82% and 3.48% without and with BN layers, respectively. The ROC curves of the two models on the handheld set are plotted in Fig. 10. The comparisons suggest that TBE-CNN effectively makes use of the complementary information between the holistic face image and facial components. It is important to note that TBE-CNN is quite efficient in both time and video memory costs. In brief, the time and memory costs of TBE-CNN are only 1.26× and 1.2× those of the trunk network, respectively. In comparison, if there were no layer sharing, the total time and memory costs of the three networks would be 1.71× and 1.6× those of the trunk network, respectively.

6.5 Performance Comparison on PaSC

We next present the best performance of TBE-CNN on PaSC with TS training data and MDR-TL fine-tuning. For fair comparison with existing approaches, we present both results of TBE-CNN with and without BN layers. Similar to the previous experiment, we only add BN layers after convolutional layers of the Inception 4 and Inception 5 modules to save video memory. Performance comparison is illustrated in Fig. 11 and Table 6. The performance of the VGG Face model [7], [52] is reported according to the model published by the authors. According to the description in [7], we fine-tune the last fully connected layer of the VGG Face model with the triplet loss on CASIA-WebFace. For VGG Face model testing, we employ the same strategy as TBE-CNN to extract video representations, as described in Section 5.

In Table 6, TBE-CNN^1 denotes the TBE-CNN performance with the original face detection results from the database. TBE-CNN^2 denotes the performance based on manually cleaned face detection results. It is clear that the proposed approach achieves the best performance on both the control and handheld sets. In particular, the proposed approach outperforms the VGG Face model [7] by 4.98% and 8.82% on the control and handheld sets, respectively. It is worth noting that the VGG Face model is trained with over 2.6 million still image data and aggressive data augmentation. In comparison, the size of our original training data is only 0.49 million.

Moreover, with the proposed techniques, we win the first place with considerable margin in the BTAS 2016 Video Person Recognition Evaluation [62], which is also based on the PaSC database. With fully automatic face detection, alignment, and recognition, our four-model ensemble system achieves 98.0% and 97.0% verification rates at 1% FAR on the two sets of PaSC, respectively. The detailed comparisons on PaSC justify the effectiveness of the proposed methods for VFR.

6.6 Performance Comparison on COX Face

The rank-1 identification rates for TBE-CNN and state-of-the-art algorithms on the COX Face database are tabulated in Tables 7 and 8. In Table 7, Vi-S (S-Vi) represents the test using the i-th video set as probe (gallery) and the still images as the gallery (probe). Similarly, in Table 8, Vi-Vj represents the test using the i-th video set as probe and the j-th video set as the gallery. For both TBE-CNN and VGG Face [7], we directly deploy the models in the previous experiment to evaluate their performance on COX Face.

Under all experimental settings, TBE-CNN achieves the best performance. It is also clear that the S2V and V2S tasks are significantly more difficult than the V2V task, suggesting a huge difference in distribution between the still image domain and video data domain. Interestingly, TBE-CNN has overwhelming performance advantage in S2V and V2S tasks. This may be because it is trained with both still image and simulated video data, meaning it can learn invariant face representations from still images and video data from the same subject. This is a great advantage for many real-world VFR tasks, e.g., watch-lists in video surveillance applications, where the gallery

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2700390, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 7
Rank-1 Identification Rates (%) under the V2S/S2V Settings for Different Methods on the COX Face Database

	V2S Identification Rate			S2V Identification Rate		
	V1-S	V2-S	V3-S	S-V1	S-V2	S-V3
PSCL [1]	38.60 ± 1.39	33.20 ± 1.77	53.26 ± 0.80	36.39 ± 1.61	30.87 ± 1.77	50.96 ± 1.44
LERM [63]	45.71 ± 2.05	42.80 ± 1.86	58.37 ± 3.31	49.07 ± 1.53	44.16 ± 0.94	63.83 ± 1.58
VGG Face [7]	88.36 ± 1.02	80.46 ± 0.76	90.93 ± 1.02	69.61 ± 1.46	68.11 ± 0.91	76.01 ± 0.71
TBE-CNN	**93.57 ± 0.65**	**93.69 ± 0.51**	**98.96 ± 0.17**	**88.24 ± 0.40**	**87.86 ± 0.85**	**95.74 ± 0.67**

TABLE 8
Rank-1 Identification Rates (%) under the V2V Setting for Different Methods on the COX Face Database

	V2-V1	V3-V1	V3-V2	V1-V2	V1-V3	V2-V3
PSCL [1]	57.70 ± 1.40	73.17 ± 1.44	67.70 ± 1.70	62.77 ± 1.02	78.26 ± 0.97	68.91 ± 2.28
LERM [63]	65.94 ± 1.97	78.24 ± 1.32	70.67 ± 1.88	64.44 ± 1.55	80.53 ± 1.36	72.96 ± 1.99
HERML-GMM [61]	95.10 ± -	96.30 ± -	94.20 ± -	92.30 ± -	95.40 ± -	94.50 ± -
VGG Face [7]	94.51 ± 0.47	95.34 ± 0.32	96.39 ± 0.42	93.39 ± 0.56	96.10 ± 0.27	96.60 ± 0.52
TBE-CNN	**98.07 ± 0.32**	**98.16 ± 0.23**	**97.93 ± 0.20**	**97.20 ± 0.26**	**99.30 ± 0.16**	**99.33 ± 0.19**

TABLE 9
Mean Verification Accuracy on the YouTube Faces Database (Restricted Protocol)

	# Crop	# Training Data (Original)	Accuracy(%) ± S_E
Deep Face* [36]	1	4.4M	91.40 ± 1.10
DeepID2+ [6]	50	0.29M	93.20 ± 0.20
FaceNet [5]	2	260M	95.12 ± 0.39
VGG Face* [7]	30	2.62M	91.60 ± -
TBE-CNN*	2	0.49M	93.84 ± 0.32
TBE-CNN	2	0.49M	**94.96 ± 0.31**

is composed of high-quality ID photos and the probe includes video frames captured by surveillance cameras. Furthermore, compared to image-set model-based VFR methods [1], [61], [63], TBE-CNN has two key advantages: first, the extracted video representation is very compact, which means it is efficient for retrieval tasks; and second, the representation is robust to fluctuations in the volume of frames in a video.

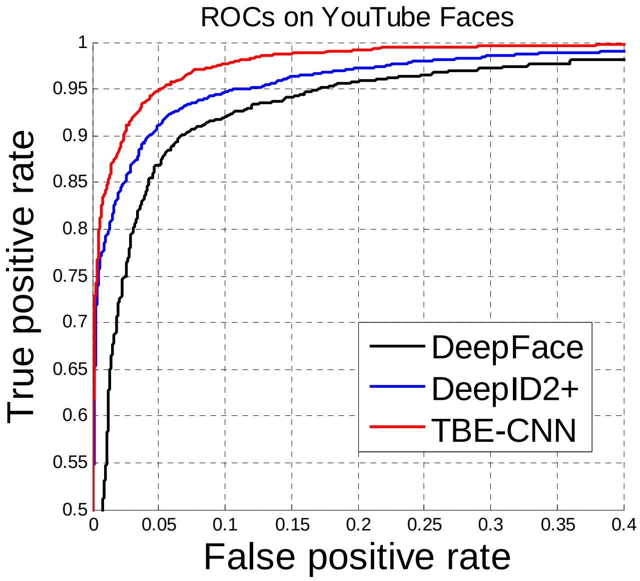


Fig. 12. ROC curves of TBE-CNN and state-of-the-art methods on the YouTube Faces database under the “restricted” protocol.

6.7 Performance Comparison on YouTube Faces

Finally, we compare the face verification performance of TBE-CNN with state-of-the-art approaches on the YouTube Faces database. As mentioned above, professional photographers rather than surveillance video cameras usually recorded videos in this database. Therefore, they are free from image blur. Also, since the majority of subjects in the video were in interviews, there is little pose variation, as illustrated in Fig. 6. Instead, the video frames are low resolution and contain serious compression artifacts. These characteristics make experiments on YouTube Faces more similar to the traditional SIFR task rather than the real-world VFR task. Therefore, existing SIFR models that are trained with large-scale still image databases, e.g., the FaceNet model [5], may be expected to show advantages in this experiment. The mean verification rates under the “restricted” protocol [13] of state-of-the-art approaches are tabulated in Table 9. Corresponding ROC curves are plotted in Fig. 12. Under this protocol, the subject identity labels of YouTube Faces database are not allowed to be used for training. In Table 9, * denotes the models without training or fine-tuning by deep metric learning methods.

TBE-CNN outperforms previous approaches such as Deep Face [36], DeepID2+ [6], and VGG Face [7]. The performance of TBE-CNN is slightly lower than the FaceNet model [5] that was trained with as many as 260

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2017.2700390, IEEE Transactions on Pattern Analysis and Machine Intelligence

million face images. Since there is no publicly available database of similar size, it will always be difficult to compete with the FaceNet model fairly. We directly cite the performance of the VGG Face model on YouTube Faces reported in [7]. TBE-CNN* outperforms VGG Face* by as much as 2.24%. This result is consistent with results on the PaSC and COX Face databases.

DeepID3 [64] is also an important recent work that was evaluated on the LFW database [3]. It is a huge ensemble system that includes 50 individual CNNs. It is also worth noting that DeepID3 and TBE-CNN are designed for different face recognition tasks and therefore have their respective advantages. The verification rate of TBE-CNN on LFW with the "Unrestricted, Labeled Outside Data Results" protocol [3] is $99.08 \pm 0.17\%$. In comparison, the best performance of an individual CNN based on GoogLeNet in DeepID3 is about 98.68%. Therefore, compared with DeepID3, TBE-CNN has its own advantage.

7 Conclusion

VFR is a challenging task due to severe image blur, rich pose variations, and occlusion. Compared to SIFR, VFR also has more demanding efficiency requirements. Here we address these problems via a series of contributions. First, to deal with image blur, we enrich the CNN training data by applying artificial blur to make up for the shortage of real-world video training data. A training set composed of still images and their blurred versions encourages CNN to learn blur-robust representations. Second, to extract pose- and occlusion-robust representations efficiently, we propose a novel CNN architecture named TBE-CNN. TBE-CNN efficiently extracts representations of the holistic face image and facial components by sharing the low- and middle-level layers of different CNNs. It improves on single CNN model performance with only marginal increases in time and memory costs. Finally, to further promote the discriminative power of the representations learnt by TBE-CNN, we propose a novel deep metric learning approach named MDR-TL, which outperforms the widely adopted triplet loss by a considerable margin. Extensive experiments have been conducted for S2V, V2S, and V2V tasks. Since the proposed TBE-CNN approach effectively handles image blur, occlusion, and pose variations, it shows clear advantages compared with state-of-the-art VFR methods on three popular video face databases.

References

[1] Z. Huang, S. Shan, R. Wang, H. Zhang, S. Lao, A. Kuerban, and X. Chen, "A benchmark and comparative study of video-based face recognition on cox face database," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5967–5981, 2015.

[2] L. Best-Rowden, B. Klare, J. Klontz, and A. K. Jain, "Video-to-video face matching: Establishing a baseline for unconstrained face recognition," in *Proc. IEEE Int. Conf. Biometrics, Theory, Appl. Syst.*, 2013, pp. 1–8.

[3] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[4] S. Biswas, G. Aggarwal, P. J. Flynn, and K. W. Bowyer, "Pose-robust recognition of low-resolution face images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 3037–3049, 2013.

[5] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 815–823.

[6] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2892–2900.

[7] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2015, pp. 1–12.

[8] H. Li and G. Hua, "Hierarchical-pep model for real-world face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4055–4064.

[9] J. R. Beveridge, H. Zhang *et al.*, "Report on the fg 2015 video person recognition evaluation," in *Proc. IEEE Int. Conf. Automatic Face and Gesture Recognit.*, 2015, pp. 1–8.

[10] J. Liu, Y. Deng, and C. Huang, "Targeting ultimate accuracy: Face recognition via deep embedding," *arXiv preprint arXiv:1506.07310*, 2015.

[11] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1988–1996.

[12] J. R. Beveridge, J. Phillips, D. S. Bolme, B. Draper, G. H. Givens, Y. M. Lui, M. N. Teli, H. Zhang, W. T. Scruggs, K. W. Bowyer *et al.*, "The challenge of face recognition from digital point-and-shoot cameras," in *Proc. IEEE Int. Conf. Biometrics, Theory, Appl. Syst.*, 2013, pp. 1–8.

[13] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 529–534.

[14] J. Phillips, J. R. Beveridge, D. S. Bolme, B. Draper, G. H. Givens, Y. M. Lui, S. Cheng, M. N. Teli, H. Zhang *et al.*, "On the existence of face quality measures," in *Proc. IEEE Int. Conf. Biometrics, Theory, Appl. Syst.*, 2013, pp. 1–8.

[15] S. Mau, S. Chen, C. Sanderson, and B. C. Lovell, "Video face matching using subset selection and clustering of probabilistic multi-region histograms," *arXiv preprint arXiv:1303.6361*, 2013.

[16] J. R. Barr, K. W. Bowyer, P. J. Flynn, and S. Biswas, "Face recognition from video: A review," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 26, no. 05, 2012.

[17] M. Bicego, E. Grosso, and M. Tistarelli, "Person authentication from video of faces: a behavioral and physiological approach using pseudo hierarchical hidden markov models," in *Advances in Biometrics*, 2006, pp. 113–120.

[18] A. Hadid and M. Pietikäinen, "Combining appearance and motion for face and gender recognition from videos," *Pattern Recognit.*, vol. 42, no. 11, pp. 2818–2827, 2009.

[19] Y.-C. Chen, V. M. Patel, P. J. Phillips, and R. Chellappa, "Dictionary-based face recognition from video," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 766–779.

[20] L. Liu, L. Zhang, H. Liu, and S. Yan, "Toward large-population face identification in unconstrained videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 11, pp. 1874–1884, 2014.

[21] Y. Hu, A. S. Mian, and R. Owens, "Face recognition using sparse approximated nearest points between image sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 1992–2004, 2012.

[22] P. Zhu, W. Zuo, L. Zhang, S. C.-K. Shiu, and D. Zhang, "Image set-based collaborative representation for face recognition," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 7, pp. 1120–1132, 2014.

[23] M. T. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell, "Graph embedding discriminant analysis on grassmannian manifolds for improved image set matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 2705–2712.

[24] Z. Cui, S. Shan, H. Zhang, S. Lao, and X. Chen, "Image sets alignment for video-based face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2626–2633.

[25] M. Shao, D. Tang, Y. Liu, and T.-K. Kim, "A comparative study of video-based object recognition from an egocentric viewpoint," *Neurocomputing*, 2015.

[26] L. Wolf, T. Hassner, and Y. Taigman, "Effective unconstrained face recognition by combining multiple descriptors and learned background statistics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 1978–1990, 2011.

[27] C. Ding and D. Tao, "A comprehensive survey on pose-invariant face recognition," *ACM Trans. Intell. Syst. Technol.*, vol. 7, pp. 37:1–37:42, 2016.

[28] M. Nishiyama, A. Hadid, H. Takeshima, J. Shotton, T. Kozakaya, and O. Yamaguchi, "Facial deblur inference using subspace analysis for recognition of blurred faces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 838–845, 2011.

[29] R. Gopalan, S. Taheri, P. Turaga, and R. Chellappa, "A blur-robust descriptor with applications to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1220–1226, 2012.

[30] C. H. Chan, M. A. Tahir, J. Kittler, and M. Pietikainen, "Multiscale local phase quantization for robust component-based face recognition using kernel fusion of multiple descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1164–1177, 2013.

[31] T. Ahonen, E. Rahtu, V. Ojansivu, and J. Heikkila, "Recognition of blurred faces using local phase quantization," in *Int. Conf. Pattern Recognit.*, 2008, pp. 1–4.

[32] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman, "A compact and discriminative face track descriptor," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1693–1700.

[33] H. Li, G. Hua, X. Shen, Z. Lin, and J. Brandt, "Eigen-pep for video face recognition," in *Proc. Asian. Conf. Comput. Vis.*, 2014.

[34] D. Chen, X. Cao, D. Wipf, F. Wen, and J. Sun, "An efficient joint formulation for bayesian face verification," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016.

[35] A. RoyChowdhury, T.-Y. Lin, S. Maji, and E. Learned-Miller, "Face identification with bilinear cnns," *arXiv preprint arXiv:1506.01342*, 2015.

[36] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1701–1708.

[37] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1875–1882.

[38] J. Lu, G. Wang, W. Deng, P. Moulin, and J. Zhou, "Multi-manifold deep metric learning for image set classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1137–1145.

[39] W. Ouyang and X. Wang, "Joint deep learning for pedestrian detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2056–2063.

[40] P.-A. Savalle, S. Tsogkas, G. Papandreou, and I. Kokkinos, "Deformable part models with cnn features," in *Proc. Europ. Conf. Comput. Vis., Parts and Attributes Workshop*, 2014.

[41] R. Girshick, F. Iandola, T. Darrell, and J. Malik, "Deformable part models are convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 437–446.

[42] L. Wan, D. Eigen, and R. Fergus, "End-to-end integration of a convolution network, deformable parts model and non-maximum suppression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 851–859.

[43] W. Yang, W. Ouyang, H. Li, and X. Wang, "End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3073–3082.

[44] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based r-cnns for fine-grained category detection," in *Proc. Europ. Conf. Comput. Vis.*, 2014, pp. 834–849.

[45] N. Zhang, E. Shelhamer, Y. Gao, and T. Darrell, "Fine-grained pose prediction, normalization, and recognition," *arXiv preprint arXiv:1511.07063*, 2015.

[46] S. Huang, Z. Xu, D. Tao, and Y. Zhang, "Part-stacked cnn for fine-grained visual categorization," *arXiv preprint arXiv:1512.08086*, 2015.

[47] S. R. Arashloo and J. Kittler, "Energy normalization for pose-invariant face recognition based on mrf model image matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1274–1280, 2011.

[48] R. Wang and D. Tao, "Recent progress in image deblurring," *arXiv preprint arXiv:1409.6838*, 2014.

[49] J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 769–777.

[50] L. Xu, J. S. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," in *Proc. Adv. Neural Inform. Process. Syst.*, 2014, pp. 1790–1798.

[51] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

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

[53] H. Elad and N. Ailon, "Deep metric learning using triplet network," in *Proc. Int. Workshop Similarity-Based Pattern Recognit.*, 2015.

[54] H. Karcher, "Riemannian center of mass and so called karcher mean," *arXiv preprint arXiv:1407.2087*, 2014.

[55] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

[56] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3476–3483.

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

[58] T. Zhang, D. Tao, X. Li, and J. Yang, "Patch alignment for dimensionality reduction," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1299–1313, 2009.

[59] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 118–126.

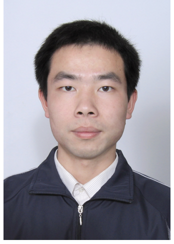
[60] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[61] Z. Huang, R. Wang, S. Shan, and X. Chen, "Face recognition on large-scale video in the wild with hybrid euclidean-and-riemannian metric learning," *Pattern Recognit.*, vol. 48, no. 10, pp. 3113–3124, 2015.

[62] J. Phillips, W. Scheirer, P. Flynn, C. Ding, G. Guo *et al.*, "Report on the btas 2016 video person recognition evaluation," in *Proc. IEEE Int. Conf. Biometrics, Theory, Appl. Syst.*, 2016, pp. 1–8.

[63] Z. Huang, R. Wang, S. Shan, and X. Chen, "Learning euclidean-to-riemannian metric for point-to-set classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1677–1684.

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



Changxing Ding received the PhD degree from the University of Technology Sydney, Australia. His research interests include computer vision, machine learning, and especially focus on face recognition.

Dacheng Tao (F'15) is Professor of Computer Science with the Centre for Artificial Intelligence, and the Faculty of Engineering and Information Technology in the University of Technology Sydney. He mainly applies statistics and mathematics to data analytics problems and his research interests spread across computer vision, data science, image processing, machine learning, and video surveillance. His research results have expounded in one monograph and 100+ publications at prestigious journals and prominent conferences, such as IEEE T-PAMI, T-NNLS, T-IP, JMLR, IJCV, NIPS, ICML, CVPR, ICCV, ECCV, AISTATS, ICDM; and ACM SIGKDD, with several best paper awards, such as the best theory/algorithm paper runner up award in IEEE ICDM07, the best student paper award in IEEE ICDM13, and the 2014 ICDM 10 Year Highest-Impact Paper Award.