Neurocomputing 329 (2019) 311–319

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

“Stream loss”: ConvNet learning for face verification using unlabeled videos in the wild

Elaheh Rashedi[a,*], Elaheh Barati[a], Matthew Nokleby[b], Xue-wen Chen[a]

[a] Department of Computer Science, Wayne State University, Detroit, MI, United States
[b] Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI, United States

ARTICLE INFO

Article history:
Received 16 January 2018
Revised 6 June 2018
Accepted 18 October 2018
Available online 24 October 2018

Communicated by Dr Zhen Lei

Keywords:
Convolutional neural network
Face verification
Loss learning
Video stream

ABSTRACT

Face recognition tasks have seen a significantly improved performance due to ConvNets. However, less attention has been given to face verification from videos. This paper makes two contributions along these lines. First, we propose a method, called *stream loss*, for learning ConvNets using unlabeled videos in the wild. Second, we present an approach for generating a face verification dataset from videos in which the labeled streams can be created automatically without human annotation intervention. Using this approach, we have assembled a widely scalable dataset, *FaceSequence*, which includes 1.5M streams capturing ∼500K individuals. Using this dataset, we trained our network to minimize the *stream loss*. The network achieves accuracy comparable to the state-of-the-art on the LFW and YTF datasets with much smaller model complexity. We also fine-tuned the network using the IJB-A dataset. The validation results show competitive accuracy compared with the best previous video face verification results.

1. Introduction

Face verification aims to determine whether two faces in a given pair of images or videos belong to the same identity or not, without having any prior knowledge about that identity. A variety of image descriptors such as SIFT [1], LBP [2,3], HOG [4], and Fisher Vector [5,6] has been proposed to be used for extracting features in face verification. However, due to variations in pose, illumination, resolution, and facial expression, face verification is still a challenging problem.

In the field of face recognition, deep learning models such as DeepFace [7], and FaceNet [8] are proven to outperform the traditional shallow methods on the widely used benchmarks such as LFW [9] and YTF [2]. In video-based face recognition, these models fall into two main branches.

In the first branch of video-based face recognition methods, a face video is represented as a set of frame-level face features as in [7,8,10]. These methods feed the video into the ConvNet as a series of selected frames and the rest of the process is similar to still-image based face recognition tasks. Although these methods have experimented on face video datasets, the temporal relationship between the frames is ignored. In other words, these methods ignore motion information in the dynamic content of videos, which can provide a promising improvement in the image recognition tasks, especially in face verification.

In the second branch of video-based face recognition methods, face verification is performed by sending the video directly to the ConvNet as an input. Although few methods such as [11] have leveraged deep ConvNets in their face recognition models, recognizing faces using deep neural networks in unconstrained videos is still in its infancy. On the one hand, the quality of video frames are significantly lower than images in the standard face image datasets, and a few ConvNet-based face recognition methods consider this characteristic of videos, i.e. motion blurred images, when extending from image to video face recognition. On the other hand, existing face video datasets are usually small in volume. Accordingly, due to lack of reliable training data in video-based face verification approaches, the ConvNets are first trained on large-scale image datasets, and then fine-tuned with existing small video datasets [12]. However, an effective approach to enhance the performance of video-based face verification is to train the model using a real-world video dataset.

In face verification methods, and more particularly in the case of using video datasets, the feature representation of each face image obtained from a ConvNet requires to be discriminative since the label prediction is not applicable while training the ConvNet. These features need to be learned using a loss function that should be computed in advance. Among different types of loss functions, one can mention contrastive loss [13–15] which constructs loss for image pairs, and triplet loss [8] which accepts a triple of images as

[*] Corresponding author.
E-mail address: elahehrashedi@gmail.com (E. Rashedi).

https://doi.org/10.1016/j.neucom.2018.10.041
0925-2312/© 2018 Elsevier B.V. All rights reserved.

the input and enhances the discriminative power of face features. Triplet loss is employed for face verification to minimize the distance between two feature vectors from the same identity; however, when the data is video, triplet loss does not take advantage of the sequence of the frames.

In this paper, we propose a new loss learning approach, entitled *stream loss*, to enhance the power of discriminative face features in ConvNets using the temporal connectivity of frames. Specifically, in addition to the original and the negative face images, we leverage hidden information in videos by importing a sequence of positive frames into the network. In other words, we account for encoded additional information in videos by using a number of sequential frames for each identity. We also approach the problem of the small volume of video training data with presenting a new real-world face video dataset for training the model. To sum up, our main contributions are as follows:

- We propose a new loss learning approach (called *stream loss*) for ConvNet training using an unlabeled video dataset. *Stream loss* achieves competitive performance comparing to the state-of-the-art in face verification while reducing the number of model parameters and training samples required by half.
- We present an automatic strategy for generating a real-world video face verification dataset from videos collected in-the-wild.
- We have assembled the *FaceSequence* dataset, which includes 1.5M streams that capture more than 500K different individuals to this end. A key distinction between this dataset and existing video datasets is that *FaceSequence* is generated from publicly available videos and labeled automatically, hence widely scalable at no annotation cost.

In the remainder of this paper, first, we provide an overview of the most related approaches in video-based face verification. Then, we introduce the proposed model, including the architectural design and *stream loss* learning method. The face retrieval approach to obtain the video stream dataset is also explained. Thereafter, we describe the training task and evaluation of the proposed model on the LFW and YTF datasets. Following that, we present the experiment of transferring the knowledge of parameters into a modified network to evaluate and compare the proposed model with state-of-the-art face verification methods. Thereafter, we provide a comparison between the generated dataset (i.e. *FaceSequence*) and other face datasets. Finally, the summary and the scope for future work is given.

2. Related work

In the recent past, many attempts have been made in face recognition algorithms based on deep learning. Existing deep learning methods are mainly introduced based on deep belief networks (DBN) [16], stacked auto-encoder [17], and convolutional neural networks (ConvNet) [18,19]. Among those, ConvNets have dramatically improved the state-of-the-art in face recognition [20].

Although ConvNet-based methods have acquired promising results in face verification, they are mostly limited to still images, rather than videos. In this work, we contribute to the second category and we propose a ConvNet-based metric learning for face verification using video streams. Here, we review the literature in two main parts, 1) ConvNet-based loss learning methods for face recognition, and 2) ConvNet-based face recognition methods for video streams.

2.1. ConvNet-based loss learning methods for face recognition

The loss functions learned by ConvNet-based face recognition methods can be categorize into three groups, 1) contrastive loss [14,21–25], 2) triplet loss [8,10,26–28], and 3) multiple loss [29–32].

In 2014, Taigman et al. [21] developed an effective deep ConvNet that combines the output of the network with PCA for dimension reduction and an SVM for classification. For verification, the model employs the Siamese network as an end-to-end method for learning a verification metric. The verification metric is defined as the L_2 distance between two feature vectors. Then, the model is trained using still face image datasets. A similar approach has been utilized in [22] and [24].

In the same year, Sun et al. [14] proposed a deep ensemble ConvNet which is trained by using a combination of classification and verification loss. The verification loss is defined as a *joint Bayesian* metric which minimizes the L_2 distance between positive face pairs, while it enforces a distance margin between negative pairs. In this method, only one pair of images are compared in each training step [33]. Likewise, in [34] and [25], the joint Bayesian loss is learned for verification using a Siamese network.

In another attempt, Hu et al. [23] introduced a deep metric learning method for face verification using ConvNets. In this method, a *Mahalanobis distance* metric is learned to minimize the distance between faces of the same identity and maximize the distance between faces of different identities. The model utilized the unrestricted still images taken from LFW imageset, as well as YTF video frames.

Later in 2015, Schroff et al. [8] presented a ConvNet model for face verification which directly learns a mapping from face images into an Euclidean space. The proposed ConvNet model learns triplet loss motivated from [35]. Triplet loss ensures that the original image of a face identity (x_o) is closer to positive examples of that identity (x_p) than it is to negatives examples (x_n). Unlike the previous methods in which only pairs of images are compared, the triplet loss enforces a relative distance constraint two pairs out of triplet images. The effectiveness of triplet loss has been demonstrated in [10] and [26] for ConvNet-based face recognition.

Following that, a generalized version of triplet loss presented in [28], named *multi-class N-pair loss*, which generalizes triplet loss by allowing joint comparison among N-1 negative examples chosen from disparate still images.

In 2016, Wen et al. [29] proposed a multiple loss function named *center loss*. In this approach, The ConvNet learns the center of each class of features and minimizes the distances between the features and their corresponding class centers. The ConvNet learning is then supervised by a combination of center loss with the softmax loss. A similar approach has being provided by COCO algorithm in 2017 [32].

In another study, Zhang et al. [30] provided a multiple loss function called *Range loss*, in which the optimization objective is to minimize the intra-class variations and enlarge the inter-class differences.

Our proposed loss learning method falls into the third category, i.e. multiple loss, where the objective is to optimize the similarity/dissimilarity of a video stream with positive/negative examples.

2.2. ConvNet-based face recognition methods for video streams

One simple approach toward adapting still-image-based ConvNet methods to videos is to represent a video as individual frames where the frame-level features are recognized individually, and then combined together to generate the video-level features [36]. However, the influence of additional temporal or dynamic information available in a sequence of frames is not considered in this recognition approach.

To the best of our knowledge, few attempts have been made on video-based face verification. Dong et al. [11] proposed an ensemble of three units network architecture called “input aggregated

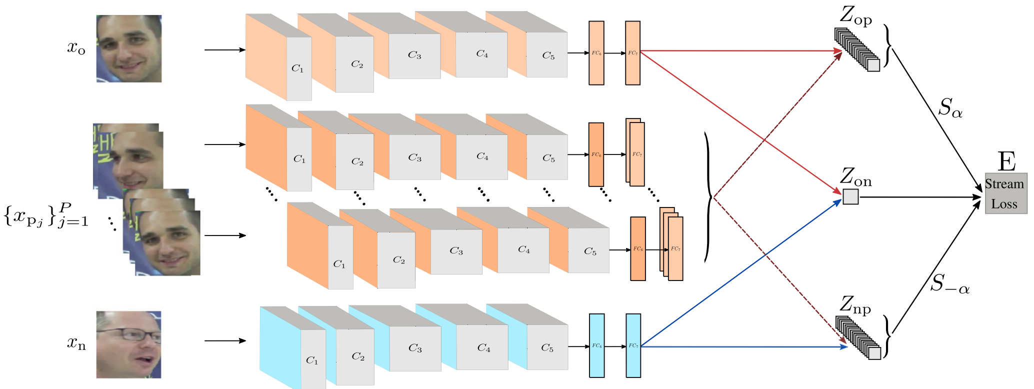

Fig. 1. The architecture of the proposed stream-based ConvNet for video face recognition. The model is composed of $(P+2)$ base ConvNets with shared parameters, joining together in a $(P+2)$-way loss layer, where P represents number of frames in the stream.

network” to identify faces in videos. This network contains a deep ConvNet as a frame representation unit and an aggregation unit in which frame features are modeled as one Riemannian manifold point. These points are mapped into high dimensional space through mapping unit.

In another study, Ding et al. [27] proposed an ensemble model for video-based face recognition, called Trunk-Branch-Ensemble-CNN model (TBE-CNN), which regularizes the triplet loss by considering the distribution of triplet samples. In [27], researchers claim that most available video datasets are rather small in volume for training video-based ConvNets. To alleviate this limitation, they simulated large amounts of video frames from existing still face image datasets and applied a random artificial blur to the streams (the authors referred to this data as “artificially simulated video data”). Then, they trained the ConvNet with the combination of the simulated streams and still face images. This method solved the problem of image blur in video-based recognition, yet it ignores the temporal evolution of the frames.

Later, Yang et al. [37] proposed a neural aggregation network, called “NAN”, which takes a face video or a set of face images as input, and produces feature representation using two modules, i) a deep ConvNet for mapping each face image into a feature vector, and ii) an aggregation block to make a single representative feature. This network uses “face image set” for training, as well as sequence of video-frames.

In the following, we explain our proposed stream-based ConvNet learning method as well as the face video dataset collection and labeling strategy.

3. Proposed stream-based ConvNet learning method

In this section, we introduce the proposed stream-based ConvNet learning method. First, we determine the architectural design of the network. Then, we introduce the *stream loss* learning approach. And thereafter, we explain the stream sampling strategy from videos.

3.1. Architecture design

The proposed model is composed of $(P+2)$ base convolutional neural networks with similar architecture and shared parameters; each base network includes 5 convolution layers and 3 fully connected layers (inspired by AlexNet [18]). The final fully-connected layers of all individual base networks meet in a $(P+2)$-way loss layer, entitled *stream loss*. The architecture of the proposed network is summarized in Fig. 1. It is worth noting that in this architecture, the $(P+2)$ copies of AlexNet are running in parallel while utilizing shared weights. Accordingly, the training time is comparable to a similar architecture with only one copy of Alexnet. Besides, the base networks can be replaced with deeper networks such as VGG ConvNet [38], to obtain higher performance with the cost of higher computation time.

In the following section, we describe the concept of *stream loss* and how it can be optimized.

3.2. Stream loss learning

In stream learning, the goal is to enforce the maximum distance between the *original example* and *positive example stream* to be comparably less than the minimum distance between the *negative example* and *positive example stream*.

Suppose that x_{o} is the original face identity, $\{x_{\mathrm{p}_j}\}_{j=1}^{P}$ is a stream of P positive examples of the original identity, and x_{n} is a negative example. Hence, for each individual identity, the network receives a set $\{x_{\mathrm{o}}, \{x_{\mathrm{p}_j}\}_{j=1}^{P}, x_{\mathrm{n}}\}$ as the input images, and generates a corresponding set $\{y_{\mathrm{o}}, \{y_{\mathrm{p}_j}\}_{j=1}^{P}, y_{\mathrm{n}}\}$ as the output feature vectors.

Accordingly, the *stream loss* function E is defined in terms of the L_2 distance between each pair of samples y_{o}, $\{y_{\mathrm{p}_j}\}_{j=1}^{P}$ and y_{n}. Therefore, we minimize the loss:

$$\mathrm{E} = \frac{1}{2K} \sum_{i=1}^{K} \mathrm{E}_i(y_{\mathrm{o}}, y_{\mathrm{p}}, y_{\mathrm{n}}), \quad (1)$$

$$\mathrm{E}_i(y_{\mathrm{o}}, y_{\mathrm{p}}, y_{\mathrm{n}}) = \Big(2\mathcal{S}_{\alpha}\Big(\big\{Z_{\mathrm{op}_{i,j}}\big\}_{j=1}^{P}\Big) - Z_{\mathrm{on}_i} - \mathcal{S}_{-\alpha}\Big(\big\{Z_{\mathrm{np}_{i,j}}\big\}_{j=1}^{P}\Big) + \beta\Big), \quad (2)$$

where K is the number of identities in each batch (note that index i represents the identity, i.e. $i = \{1 \ldots K\}$, and index j represents the positive examples of that individual identity, i.e. $j = \{1 \ldots P\}$), β is a margin that is enforced between positive and negative streams, and Z_{op}, Z_{on}, Z_{np} are the L_2 Norm distance between each pair of samples y_{o}, y_{p} and y_{n}, which are formulated as below:

$$Z_{\mathrm{op}_{i,j}} = \left\| y_{\mathrm{o}_i} - y_{\mathrm{p}_{i,j}} \right\|_2^2, \quad (3)$$

$$Z_{\mathrm{on}_i} = \| y_{\mathrm{o}_i} - y_{\mathrm{n}_i} \|_2^2, \quad (4)$$

$$Z_{\mathrm{np}_{i,j}} = \left\| y_{\mathrm{p}_{i,j}} - y_{\mathrm{n}_i} \right\|_2^2, \quad (5)$$

and $\mathcal{S}_{\alpha}$ and $\mathcal{S}_{-\alpha}$ are correspondingly the smooth-max and smooth-min function, which are differentiable approximation to the maximum and minimum function. $\mathcal{S}_{\alpha}$ and $\mathcal{S}_{-\alpha}$ are calculated as below:

$$\mathcal{S}_\alpha\left(\left\{Z_{\mathrm{op}_{i,j}}\right\}_{j=1}^{P}\right) = \frac{\sum_{j=1}^{P}\left(Z_{\mathrm{op}_{i,j}} \mathrm{e}^{\alpha Z_{\mathrm{op}_{i,j}}}\right)}{\sum_{j=1}^{P} \mathrm{e}^{\alpha Z_{\mathrm{op}_{i,j}}}}, \tag{6}$$

$$\mathcal{S}_{-\alpha}\left(\left\{Z_{\mathrm{np}_{i,j}}\right\}_{j=1}^{P}\right) = \frac{\sum_{j=1}^{P}\left(Z_{\mathrm{np}_{i,j}} \mathrm{e}^{-\alpha Z_{\mathrm{np}_{i,j}}}\right)}{\sum_{j=1}^{P} \mathrm{e}^{-\alpha Z_{\mathrm{np}_{i,j}}}}, \tag{7}$$

in which α is a large positive value (in this experiment $\alpha = 1000$). Here $\mathcal{S}_\alpha$ approximates the maximum distance between the original example y_{o} and all P positive examples $\{y_{\mathrm{p}_j}\}_{j=1}^{P}$. Similarly, $\mathcal{S}_{-\alpha}$ approximates the minimum distance between the negative example y_{n} and all P positive examples $\{y_{\mathrm{p}_j}\}_{j=1}^{P}$.

We train the network using the standard backpropagation algorithm, in which the value of E is calculated in the forward pass and the gradients of E are calculated and propagated backward in order to update the model parameters. To do so, we calculate the partial derivatives of E, denoted by $\frac{\partial \mathrm{E}}{\partial y_{\mathrm{o}}}$, $\frac{\partial \mathrm{E}}{\partial y_{\mathrm{p}}}$ and $\frac{\partial \mathrm{E}}{\partial y_{\mathrm{n}}}$, as follows:

$$\frac{\partial \mathrm{E}}{\partial y_{\mathrm{o}_i}} = \sum_{j=1}^{P}\left(2\frac{\partial S_\alpha}{\partial Z_{\mathrm{op}_{i,j}}} \times \frac{\partial Z_{\mathrm{op}_{i,j}}}{\partial y_{\mathrm{o}_i}} - \frac{\partial Z_{\mathrm{on}_i}}{\partial y_{\mathrm{o}_i}} - \frac{\partial S_{-\alpha}}{\partial Z_{\mathrm{np}_{i,j}}} \times \frac{\partial Z_{\mathrm{np}_{i,j}}}{\partial y_{\mathrm{o}_i}}\right), \tag{8}$$

$$\left\{\frac{\partial \mathrm{E}}{\partial y_{\mathrm{p}_{i,j}}}\right\}_{j=1}^{P} = \left(2\frac{\partial S_\alpha}{\partial Z_{\mathrm{op}_{i,j}}} \times \frac{\partial Z_{\mathrm{op}_{i,j}}}{\partial y_{\mathrm{p}_{i,j}}} - \frac{\partial S_{-\alpha}}{\partial Z_{\mathrm{np}_{i,j}}} \times \frac{\partial Z_{\mathrm{np}_{i,j}}}{\partial y_{\mathrm{p}_{i,j}}}\right), \tag{9}$$

$$\frac{\partial \mathrm{E}}{\partial y_{\mathrm{n}_i}} = \sum_{j=1}^{P}\left(2\frac{\partial S_\alpha}{\partial Z_{\mathrm{op}_{i,j}}} \times \frac{\partial Z_{\mathrm{op}_{i,j}}}{\partial y_{\mathrm{n}_i}} - \frac{\partial Z_{\mathrm{on}_i}}{\partial y_{\mathrm{n}_i}} - \frac{\partial S_{-\alpha}}{\partial Z_{\mathrm{np}_{i,j}}} \times \frac{\partial Z_{\mathrm{np}_{i,j}}}{\partial y_{\mathrm{n}_i}}\right). \tag{10}$$

Since $\frac{\partial Z_{\mathrm{op}_{i,j}}}{\partial y_{\mathrm{n}_i}}$ and $\frac{\partial Z_{\mathrm{np}_{i,j}}}{\partial y_{\mathrm{o}_i}}$ are equal to zero, we have:

$$\frac{\partial \mathrm{E}}{\partial y_{\mathrm{o}_i}} = \sum_{j=1}^{P}\left(2\frac{\partial S_\alpha}{\partial Z_{\mathrm{op}_{i,j}}} \times \frac{\partial Z_{\mathrm{op}_{i,j}}}{\partial y_{\mathrm{o}_i}} - (y_{\mathrm{o}_i} - y_{\mathrm{n}_i})\right), \tag{11}$$

$$\frac{\partial \mathrm{E}}{\partial y_{\mathrm{n}_i}} = \sum_{j=1}^{P}\left((y_{\mathrm{o}_i} - y_{\mathrm{n}_i}) - \frac{\partial S_{-\alpha}}{\partial Z_{\mathrm{np}_{i,j}}} \times \frac{\partial Z_{\mathrm{np}_{i,j}}}{\partial y_{\mathrm{n}_i}}\right), \tag{12}$$

where the gradient terms are defined as below:

$$\frac{\partial S_\alpha\left(\left\{Z_{\mathrm{op}_{i,j}}\right\}_{j=1}^{P}\right)}{\partial Z_{\mathrm{op}_{i,j}}} = \frac{\mathrm{e}^{\alpha Z_{\mathrm{op}_{i,j}}}}{\sum_{k=1}^{P} \mathrm{e}^{\alpha Z_{\mathrm{op}_{i,k}}}}\left[1 + \alpha\,(Z_{\mathrm{op}_{i,j}} - S_\alpha\left(\left\{Z_{\mathrm{op}_{i,j}}\right\}_{i=1}^{P}\right)\right], \tag{13}$$

$$\frac{\partial S_{-\alpha}\left(\left\{Z_{\mathrm{np}_{i,j}}\right\}_{j=1}^{P}\right)}{\partial Z_{\mathrm{np}_{i,j}}} = \frac{\mathrm{e}^{-\alpha Z_{\mathrm{np}_{i,j}}}}{\sum_{k=1}^{P} \mathrm{e}^{-\alpha Z_{\mathrm{np}_{i,k}}}}\left[1 - \alpha\,(Z_{\mathrm{np}_{i,j}} - S_{-\alpha}\left(\left\{Z_{\mathrm{np}_{i,j}}\right\}_{i=1}^{P}\right)\right], \tag{14}$$

$$\frac{\partial Z_{\mathrm{op}_{i,j}}}{\partial y_{\mathrm{o}_i}} = -\frac{\partial Z_{\mathrm{op}_{i,j}}}{\partial y_{\mathrm{p}_i}} = y_{\mathrm{o}_i} - y_{\mathrm{p}_{i,j}}, \tag{15}$$

$$\frac{\partial Z_{\mathrm{np}_{i,j}}}{\partial y_{\mathrm{p}_i}} = -\frac{\partial Z_{\mathrm{np}_{i,j}}}{\partial y_{\mathrm{n}_i}} = y_{\mathrm{p}_{i,j}} - y_{\mathrm{n}_i}. \tag{16}$$

For each set of original examples, positive streams, and negative examples, we carry out a single backpropagation step.

The proposed learning method provides three advantages tailored to learning from videos, which distinguish it from triplet selection:

1. In triplet loss, the distance between the positive example and negative example is ignored, while in *stream loss* this distance is maximized.
2. In triplet loss, the hard-negative exemplars are selected from within a mini-batch, while in *stream loss* the negative samples are effectively chosen from the same video, with likely same video quality, lighting condition and matching background.
3. In triplet loss, each anchor is paired with all positive samples in a mini-batch, while in *stream loss* the same face is picked from different frames in the sequence, with same identity and varying poses. In [8] it is mentioned that correct sample selection is important for fast convergence.

3.3. Stream sample collection

One of our goals is to create the input streams $\{x_{\mathrm{o}}, \{x_{\mathrm{p}_j}\}_{j=1}^{P}, x_{\mathrm{n}}\}$ in an automatic manner. This approach improves the learning performance by avoiding the data labeling effort. Therefore, we propose the following strategy to generate the video stream dataset named *FaceStream*, inspired by VGG's dataset collection process [10].

The first stage of generating this dataset is to obtain a list of video URLs. The initial list containing random video URLs is obtained by employing web crawlers. The second stage is to manually recognize and select the videos which demonstrate human faces, and to add them to a candidate list. This stage is repeated until the candidate list of 500K video URLs is provided. The candidate videos are curated to control biases in ethnicity, gender, age, and pose varieties. The next stage is to select the original target examples, positive streams, and negative examples from each video, which is explained in the following paragraph:

Original target selection x_{o}: Each original target face is selected from a random frame of a video by employing the deep-learning-based face detection algorithm presented in [39].

Negative sample selection x_{n}: After selecting the original target, the face detection algorithm continues to detect other faces which exist in the same frame as the target. One can claim that with high probability the mentioned faces have a different identity from the target face. In the case that more than one negative sample is detected, one examples is chosen randomly. The streams with no negative samples are discarded and not further processed. The motivation behind choosing the negative example from the same frame as the target is that the identities in the same frame are mostly affected by the similar conditions such as illumination, and resolution. Moreover, two faces that appear in the same frame are more likely to have matching backgrounds. Accordingly, the dissimilarity of these two examples is less dependent on the differences in their backgrounds.

Positive sample stream selection x_{p}: The last step is to select a stream of faces from a sequence of frames in the video with the same identity as the target, whilst they still have some variation in pose, shape, illumination, etc. Here, we deploy a face tracking algorithm to track the target face in the same video for a specific time period and select the tracked faces result as a positive stream. In this experiment, we utilize the idea of the long term tracking method presented in [40]. The target is tracked within the next P consequent frames. In this strategy, P is set to 19 sequences of positive frames. In order to discuss the effect of P value on performance, it's worth mentioning that the frame-rate of the collected videos varies from 19 to 23 frames per second. Thus, we set the frame sequence length to the minimum frame rate, i.e. 19, to present ∼1 sec movement of the face in the video. Therefore, a P much smaller than 19 corresponds to small pose variations, which is unlikely to provide enough dissimilarity between frames. Accordingly, we expect lower performance for reduced P.

The original, negative and positive stream examples are assembled in a dataset named *FaceSequence*. An example of five streams available in this dataset is provided in Fig. 2. As it is illustrated, for each identity x_{o} in the *FaceSequence* dataset, there exists one

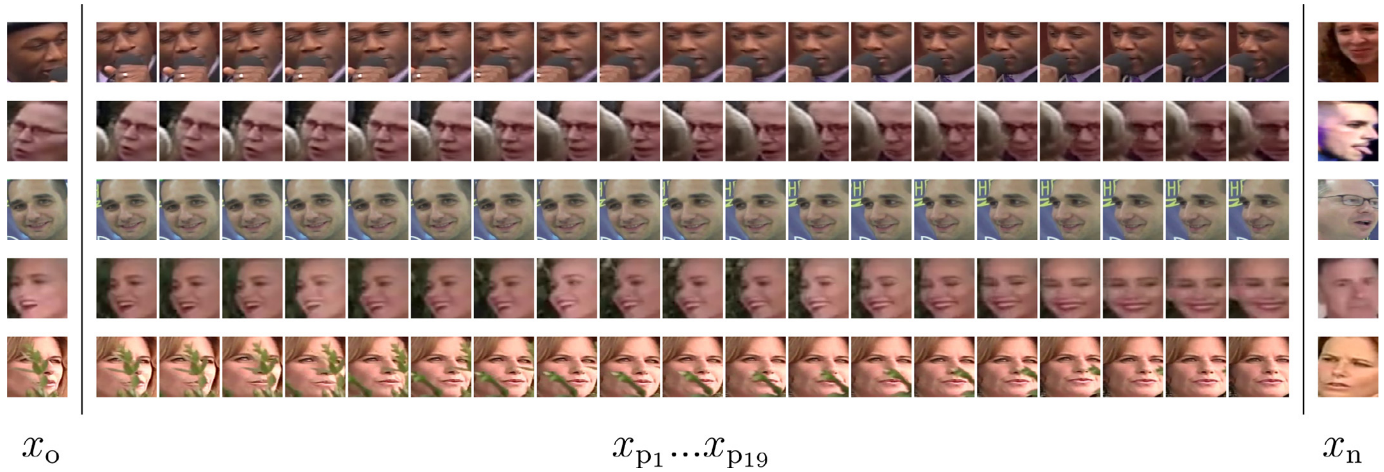

Fig. 2. An example of stream of frames available in *FaceSequence* dataset for 5 identities. The first column includes the original example x_o, the last column includes the negative example x_n, and the middle columns indicate the stream of positive examples $\{x_{p_j}\}_{j=1}^{19}$.

Table 1
Characteristics of the *FaceSequence* dataset, including total number of videos, number of streams extracted from videos, and number of frames per each stream.

Dataset	FaceSequece
# Videos	500K
# Streams	1.5M
# Frames-per-stream	21[a]

[a] Here, P is set to 19 sequences of positive frames. Accordingly, the length of the stream including x_o, $\{x_{p_j}\}_{j=1}^{19}$ and x_n is $P+2=21$.

positive stream $\{x_{p_j}\}_{j=1}^{19}$ (including 19 consequent frames), and one negative example x_n. At the end, 1.5M number of streams are collected from 500K videos, with an average of 3 identities per video. Table 1 shows the statistics of the *FaceSequence* dataset.

4. Experimental results

Here, we evaluate the performance of the proposed model (stream-based ConvNet) using two different protocols. First we follow the protocol of *unrestricted with labeled outside data* and test our model on both *still* and *video* datasets, LFW and YTF. Then we fine-tune our pre-trained model on the IJB-A video dataset [41]. Thereafter, we provide the results of face verification task on the IJB-A dataset. Finally, we present a comparison between the generated dataset (i.e. *FaceSequence*) and other face datasets.

4.1. Experiments on LFW and YTF datasets

We evaluate our method for face verification by using two famous face datasets in unconstrained environments, LFW [9] and YTF [2]. The LFW dataset includes 13,233 images from 5749 different identities, which is a standard dataset for evaluating the face recognition tasks such as face verification. The YTF dataset, as a standard benchmark for unconstrained face verification in videos, includes 3425 videos from 1595 different identities. The length of video clips in YTF dataset ranges from 48 to 6070 frames, with the average of 181.3 frames per video. Our model is trained on 1.5M face streams from the generated *FaceSequence* dataset with no identity overlapping with LFW and YTF.

4.1.1. Verification evaluation

We followed the evaluation procedure as defined in [8]. In face verification, we are given a pair of face images $\{x_o, x_u\}$, where x_o is the original image, and x_u is the identity to be verified. The network maps the input pair to a feature space of $\{y_o, y_u\}$. Accordingly, the L_2 distance for the given input pair is defined as:

Table 2
Comparison of verification performance of different methods on the LFW and YTF datasets.

Method	# Images	# Networks	Acc. on LFW	Acc. on YTF
DDML (combined) [23]	–	1	90.68%	82.3%
WebFace+PCA [34]	500K	1	96.33%	90.6%
DeepFace [7]	4M	3	97.35%	91.4%
DeepID2+ [24]	300K	25	99.47%	93.2%
MFM 2/1 [42]	–	1	98.8%	93.4%
RangeLoss [30]	1.5M	–	99.52%	93.7%
CenterLoss [29]	0.7M	1	99.28%	94.9%
FaceNet [8]	200M	3	99.63%	95.1%
VGG [10]	2.6M	3	98.95%	97.3%
Baidu [26]	1.3M	1	99.13%	–
NAN [37]	3M	1	–	95.7%
TBE-CNN [27]	–	1	–	94.96%
COCO [32]	–	1	99.78%	–
SphereFace [43]	500K	1	97.88%–99.42%	93.1%–95.0%
NormFace [44]	500K	10	98.13%–98.71%	94.72
Stream loss	1.5M[a]	21	98.97%	96.4%

[a] Here, the the dataset includes 1.5M streams.

$$\mathcal{D}_{(a_o,a_u)} = \|y_o - y_u\|^2 \tag{17}$$

Where $\mathcal{D}_{(x_o,x_u)}$ is utilized to determine the classification of *different* identities (class:0) or the *same* identity (class:1) (see Eq. (18)):

$$C_{(x_o,x_u)} = \begin{cases} 0 & \text{if } \mathcal{D}_{(a_o,a_u)} \leq d \\ 1 & \text{otherwise.} \end{cases} \tag{18}$$

where d is the distance threshold, which is set to 0.7, this value has been chosen based on practical experiments, and some other models (e.g. FaceNet) has been used the similar value as the threshold for verification.

Following the mentioned verification strategy, we test our model on 6K face pairs in the LFW dataset, and 5K video pairs from the YTF dataset and report the accuracy of the results in Table 2.

From Table 2, it can be observed that the *stream loss* model achieves a VGG level accuracy with 2 × fewer parameters (60M vs 140M parameters) and 2 × fewer training data (1.5M data vs. 2.6M data) compared to VGG model. This demonstrates the effectiveness of *stream loss*. Although the 1.5M streams contain 30M images, nevertheless, the images in each stream are stills from 1sec of video, and hence are rather similar. For example, FaceNet and VGG are trained on rather disparate stills, thus having more information per image, while the proposed method effectively choose only three still per stream. Therefore, we contend that the number of streams, rather than the number of images, is the correct figure of merit.

In Table 2, We also provide an accuracy comparison between *stream loss* and COCO [32] on LFW and YTF. We highlight that the performance of COCO on YTF is not reported. The COCO method achieves higher accuracy on LFW, but it is difficult to conjecture the performance beyond this dataset, since most algorithms perform quite well, including CenterLoss, which has comparable performance to COCO. CenterLoss and COCO are similar in spirit, and both methods make inter-class features discriminative and use the idea of a class centroid for metric learning [32]. On the YTF dataset, *stream loss* and VGG both outperform CenterLoss. We expect a similar result from COCO.

4.2. Experiments on IJB-A video dataset

4.2.1. Transferring knowledge of parameters

Suppose that our face dataset is denoted as D_S and the verification learning task is indicated as $\mathcal{T}_S$. We aim to transfer the learning from domain D_S to a target domain D_T to perform the learning task $\mathcal{T}_T$ to improve the learning of the target prediction function. In this study, the target domain D_T is the IJB-A video dataset [41] for face recognition, and the $\mathcal{T}_T$ is the identification (classification) task performed on subjects detected from videos.

One approach towards transfer learning is to share the knowledge of parameters [45,46]. In this experiment, we transfer the weight parameters from the source trained model to a new classification model. Let's assume that the weight parameters of our source model and target model are w_S and w_T respectively, therefore,

$$w_T = w_S + v_T \tag{19}$$

where v_T is a specific set of parameters of the target task, i.e. face verification.

In order to elaborate transferring the parameters' knowledge, we fine-tuned the pre-trained network on the IARPA Janus Benchmark A (IJB-A) dataset [41]. The IJB-A dataset includes real world unconstrained image and video faces with 5397 images and 2042 videos from 500 subjects, with an average of 11.4 images and 4.2 videos per identity. Since, the identities in the IJB-A dataset come with significant variation in pose, illumination, expression, resolution, and occlusion which makes face recognition very challenging.

Here, 333 identities are randomly sampled as the training set, and the remaining 167 identities are placed in the testing set for evaluation. The results are discussed in the following section.

4.2.2. Verification evaluation

We evaluate our method by fine-tuning the proposed model on the IJB-A dataset [41] for face verification task. In verification evaluation procedure utilized Siamese network and Cosine similarity joint with Softmax. The flowchart of the mentioned procedure is shown in Fig. 3.

In the proposed procedure, the pre-trained network generates feature representations for each of input face image pairs x_o and x_u. Then the Cosine similarity between the two vectors y_o and y_u is computed as $cosin(y_o, y_u)$. This additional feature is concatenated along with the two feature vectors. Then the output of hidden layer is passed to a softmax layer which expresses if the given pair belongs to the same identity or not.

Following this procedure, we calculate the True Acceptance Rate (TAR) and False Acceptance Rate (FAR). The results are demonstrated in Table 3. In the verification task, the TAR of our method at FAR = 0.001 is 0.871 which reduces the error of Template-Adaptation [47] and NAN-cascaded-attention [37] by about 21% and 8% respectfully. In FAR = 0.01, our method reduces the error of NAN-cascaded-attention by about 6%. Note that the proposed model needs fewer parameters and training samples compare to Template-Adaptation and NAN-cascaded-attention methods (see Table 3). In addition to the speed gain, fewer parameters reduce the sample complexity of the network, which explains the near state-of-the-art performance with fewer data streams.

Table 3
Performance comparison on the IJB-A dataset. TAR/FAR: True/False Acceptance Rate for verification.

Method	# Params	1:1 Verification TAR	
		FAR = 0.001	FAR = 0.01
CNN+AvgPool [37]	140M	0.771	0.913
VGG [10]	140M	–	0.805
Template-Adaptation [47]	40M	0.836	0.939
NAN-cascaded-attention [37]	140M	0.860	0.933
Stream loss	60M	0.871	0.937

In Table 3, the only model that uses triplet loss is VGG which has been fine-tuned using triplet loss. The TAR of our method at FAR = 0.01 reduces the error of VGG by 67% which demonstrates a significant improvement. Furthermore, the VGG dataset is purified and includes a small label noise, where the labels are used later for fine-tuning the VGG network. The stream loss dataset (i.e. *FaceSequence*), by contrast, is automatically labeled and thus noisy. Therefore, stream loss is at a disadvantage compared to VGG, yet it has comparable performance.

4.3. Comparison of FaceSequence *to other face datasets*

In early face recognition datasets the main focus was to collect stills from subjects under controlled conditions such as lighting, pose, or facial expression, and hence less individuals, e.g. Yale-B [48]. In recent datasets the focus moved to collect photos of large number of individuals, and hence uncontrolled scenarios per each individual, e.g. IJB-A [41], MegaFace [49], CASIA [34]. While, the *FaceSequence* have the advantage of both. In *FaceSequence*, on the one hand, each subject's environment is relatively static in terms of background, lighting, resolution, etc. On the other hand, the images are assembled from ordinary people extracted from vast variety of videos crawled from the web and publicly available at no cost, which makes it easily scalable to millions of individuals.

In this work, we have assembled the *FaceSequence* dataset, which includes 1.5M streams that capture more than 500K different individuals to this end. Our key objectives for assembling the dataset are that:

1. *FaceSequence* contains photo streams extracted from *videos in the wild*, under variety of unconstrained conditions including resolution, pose, expression, lighting, exposure, and blurriness.
2. The images in each stream are stills from ∼1 second of video, and hence *more similar in terms of background, lighting, and resolution per subject*, comparing to common still image datasets which include many disparate stills images per individual.
3. And most importantly, it is *widely scalable*. Most public face datasets have leveraged labeled celebrity photos crawled from the web, which makes it very challenging to assemble millions of individuals. Private datasets on the other hand, are scalable by involving human annotators which makes the process costly and much more time consuming. Whilst, in *FaceSequence*, the streams are automatically labeled with no human interaction in the loop which makes it expandable.

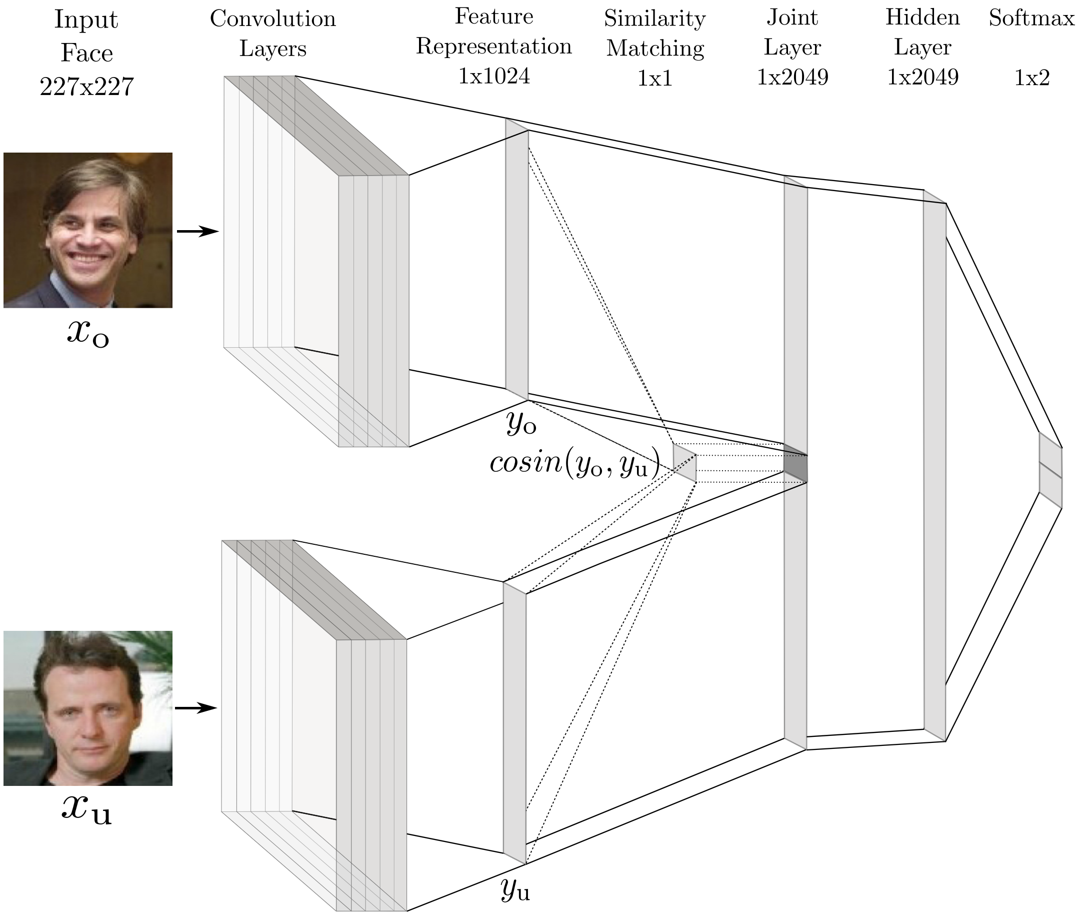

Fig. 3. The proposed network for face verification. For a given pair of $\{x_\mathrm{o}, x_\mathrm{u}\}$, we map them to a feature space of $\{y_\mathrm{o}, y_\mathrm{u}\}$. The two feature spaces are concatenated in a joint layer with an additional feature vector of size 1×1. The output of the joint layer is utilized in a softmax layer to determine whether the input pairs belong to the same identity or not.

FaceSequence will be publicly available[1], to enable benchmarking and encourage development of video-based face verification algorithms at scale.

5. Summary & future work

In this paper, a stream-based ConvNet architecture is presented for video face verification task. The proposed network is trained to optimize the differentiable error function, referred to as *stream loss*, using unlabeled temporal face sequences. In addition, a novel method for generating training dataset from videos (named *FaceSequence*) is presented based on long-term face tracking. Our method achieved comparable accuracy results on LFW and YTF datasets. Experiments on the large scale face benchmark IJB-A also demonstrate the effectiveness of the proposed *stream loss* function. For example, in comparison to VGG, our method demonstrates a significant improvement in TAR/FAR, considering the fact that the VGG dataset is highly purified and includes a small label noise.

In this experiment, we have curated the dataset to eliminate the noise. Recently, different designs of noise adaptation layer have been proposed for deep networks [50–53]. For future work, we will focus on incorporating a modification signal to the *stream loss* function to calculate the statistics of label noise. By changing the loss function, we can make the learning more robust in case of using a more noisy training data.

We will also look into different approaches for feeding streams of negative examples into ConvNet (instead of only one negative example) to improve the loss learning procedure. The *stream loss* function has to be re-designed accordingly.

Acknowledgment

This work is supported in part by the US National Science Foundation award to XWC (IIS-1554264).

References

[1] J. Sivic, M. Everingham, A. Zisserman, "who are you?"-learning person specific classifiers from video, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2009, pp. 1145–1152.
[2] L. Wolf, T. Hassner, I. Maoz, Face recognition in unconstrained videos with matched background similarity, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 529–534.
[3] C. Lu, X. Tang, Surpassing human-level face verification performance on LFW with gaussianface., in: Proceedings of the AAAI, 2015, pp. 3811–3819.
[4] M. Everingham, J. Sivic, A. Zisserman, Taking the bite out of automated naming of characters in tv video, Image Vis. Comput. 27 (5) (2009) 545–559.
[5] K. Simonyan, O.M. Parkhi, A. Vedaldi, A. Zisserman, Fisher vector faces in the wild., in: Proceedings of the BMVC, 2, 2013, p. 4.
[6] M. Guillaumin, J. Verbeek, C. Schmid, Is that you? Metric learning approaches for face identification, in: Proceedings of the IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 498–505.
[7] L. Wolf, Deepface: closing the gap to human-level performance in face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[1] A sample set of the FaceSequence dataset and the dataset generation code are anonymously available at: https://www.dropbox.com/sh/am32t666p7nzfpc/AAB2oJvytcWQtp3ObHWvId8Fa?dl=0.

[8] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: a unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.

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

[10] O.M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, in: Proceedings of the British Machine Vision Conference, 1, 2015, p. 6.

[11] Z. Dong, S. Jia, C. Zhang, M. Pei, Input aggregated network for face video representation, arXiv:1603.06655 (2016).

[12] J.R. Beveridge, H. Zhang, B.A. Draper, P.J. Flynn, Z. Feng, P. Huber, J. Kittler, Z. Huang, S. Li, Y. Li, et al., Report on the FG 2015 video person recognition evaluation, in: Proceedings of the 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), 1, IEEE, 2015, pp. 1–8.

[13] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2, IEEE, 2006, pp. 1735–1742.

[14] Y. Sun, Y. Chen, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, in: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 1988–1996.

[15] Y. Wen, Z. Li, Y. Qiao, Latent factor guided convolutional neural networks for age-invariant face recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4893–4901.

[16] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.

[17] Q.V. Le, W.Y. Zou, S.Y. Yeung, A.Y. Ng, Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 3361–3368.

[18] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[20] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[21] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: closing the gap to human-level performance in face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701–1708.

[22] Z. Zhu, P. Luo, X. Wang, X. Tang, Recover canonicalview faces in the wild with deep neural networks. CoRR, abs/1404.3543 2014, 2.

[23] J. Hu, J. Lu, Y.-P. Tan, Discriminative deep metric learning for face verification in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1875–1882.

[24] Y. Sun, X. Wang, X. Tang, Deeply learned face representations are sparse, selective, and robust, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2892–2900.

[25] J.-C. Chen, V.M. Patel, R. Chellappa, Unconstrained face verification using deep cnn features, in: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2016, pp. 1–9.

[26] J. Liu, Y. Deng, T. Bai, Z. Wei, C. Huang, Targeting ultimate accuracy: Face recognition via deep embedding, arXiv:1506.07310 (2015).

[27] C. Ding, D. Tao, Trunk-branch ensemble convolutional neural networks for video-based face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2018) 1002–1014.

[28] K. Sohn, Improved deep metric learning with multi-class n-pair loss objective, in: Proceedings of the Advances in Neural Information Processing Systems, 2016, pp. 1857–1865.

[29] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 499–515.

[30] X. Zhang, Z. Fang, Y. Wen, Z. Li, Y. Qiao, Range loss for deep face recognition with long-tailed training data, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5409–5418.

[31] C. Huang, C.C. Loy, X. Tang, Local similarity-aware deep feature embedding, in: Proceedings of the Advances in Neural Information Processing Systems, 2016, pp. 1262–1270.

[32] Y. Liu, H. Li, X. Wang, Rethinking feature discrimination and polymerization for large-scale recognition, arXiv:1710.00870 (2017).

[33] D. Chen, X. Cao, L. Wang, F. Wen, J. Sun, Bayesian face revisited: A joint formulation, in: Proceedings of the European Conference on Computer Vision, Springer, 2012, pp. 566–579.

[34] D. Yi, Z. Lei, S. Liao, S.Z. Li, Learning face representation from scratch, arXiv:1411.7923 (2014).

[35] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, Y. Wu, Learning fine-grained image similarity with deep ranking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1386–1393.

[36] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, G. Toderici, Beyond short snippets: Deep networks for video classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4694–4702.

[37] J. Yang, P. Ren, D. Chen, F. Wen, H. Li, G. Hua, Neural aggregation network for video face recognition, arXiv:1603.05474 (2016).

[38] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556 (2014).

[39] H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, A convolutional neural network cascade for face detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5325–5334.

[40] K. Zhang, E. Rashedi, E. Barati, X. Chen, Long-term face tracking in the wild using deep learning, in: Proceedings of the ACM SIGKDD Workshop on Large-scale Deep Learning for Data Mining, 2016.

[41] B.F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, A.K. Jain, Pushing the frontiers of unconstrained face detection and recognition: IARPA janus benchmark A, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1931–1939.

[42] X. Wu, R. He, Z. Sun, T. Tan, A light CNN for deep face representation with noisy labels, IEEE Trans. Inf. Forensics Secur. 13 (11) (2018) 2884–2896.

[43] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, L. Song, Sphereface: Deep hypersphere embedding for face recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1, 2017.

[44] F. Wang, X. Xiang, J. Cheng, A.L. Yuille, Normface: L2 hypersphere embedding for face verification, in: Proceedings of the 25th ACM international conference on Multimedia. ACM, 2017.

[45] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.

[46] K. Weiss, T.M. Khoshgoftaar, D. Wang, A survey of transfer learning, J. Big Data 3 (1) (2016) 1–40.

[47] N. Crosswhite, J. Byrne, C. Stauffer, O. Parkhi, Q. Cao, A. Zisserman, Template adaptation for face verification and identification, in: Proceedings of the 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG), IEEE, 2017, pp. 1–8.

[48] K.-C. Lee, J. Ho, D.J. Kriegman, Acquiring linear subspaces for face recognition under variable lighting, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 684–698.

[49] I. Kemelmacher-Shlizerman, S.M. Seitz, D. Miller, E. Brossard, The megaface benchmark: 1 million faces for recognition at scale, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4873–4882.

[50] B. Chen, W. Deng, Weakly-supervised deep self-learning for face recognition, in: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2016, pp. 1–6.

[51] I. Jindal, M. Nokleby, X. Chen, Learning deep networks from noisy labels with dropout regularization, in: Proceedings of the IEEE 16th International Conference on Data Mining (ICDM), IEEE, 2016, pp. 967–972.

[52] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, S. Belongie, Learning from noisy large-scale datasets with minimal supervision, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2017.

[53] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, R. Fergus, Training convolutional networks with noisy labels, arXiv:1406.2080 (2014).

Elaheh Rashedi is a Ph.D. candidate in the Department of Computer Science at Wayne State University. She received her Master of Science in Electrical and Computer Engineering in 2011 from Isfahan University of Technology, and her Bachelor of Science in Electrical and Computer Engineering in 2008 from University of Tehran. She is a student member of IEEE and ACM. Currently, she is working as a graduate research assistant in Data Sciences and Analytics Lab (DSAL) in the Department of Computer Science at Wayne State University, and her main research interest includes Data Sciences and Advanced Analytics, Machine Learning, Deep Learning, Biomedical Imaging, and Parallel Computation.

Elaheh Barati received her M.Sc. degree from Isfahan University of Technology. She is currently a Ph.D. candidate in computer science at Wayne State University, Detroit, Michigan. She is a student member of the ACM, and her research interests include deep learning, computer vision, big data, artificial intelligence, and reinforcement learning.

Matthew Nokleby is an assistant professor in the Department of Electrical and Computer Engineering at Wayne State University. He directs the Information Processing Lab, where they study information theory and signal processing with emphasis on machine learning, wireless communication, and sensorfusion in wireless networks.

Xue-wen Chen is a Professor of Computer Science and the founding director of Data Sciences and Analytics Lab (DSAL) in the Department of Computer Science at Wayne State University. He served as the Department Chair between 2012 and 2014. Before joining Wayne State in 2012, he was a professor in the Electrical Engineering and Computer Science Department, University of Kansas. Dr. Chen received his PhD in 2001 from Carnegie Mellon University. He is the recipient of the US National Science Foundation CAREER Award. Dr. Chen has published over 100 peer-reviewed papers in these research fields at top journals and conferences such as KDD, ICML, Bioinformatics, and IEEE TKDE. His research is funded by several federal agencies such as the National Science Foundation, National Institutes of Health, as well as some local industry. He serves as an Editorial Board Member for several international journals such as BMC Systems Biology and IEEE Access. He also served as a Conference Chair or Program Chair for several international conferences such as the Thirteen International Conference on Machine Learning and Applications (ICMLA) in 2014, the 21st ACM Conference on Information and Knowledge Management (CIKM) in 2012, and the IEEE International Conference on Bioinformatics and Biomedicine (BIBM) in 2009. He has also served as a Program Committee Member for numerous international conferences.