Neural Aggregation Network for Video Face Recognition

Jiaolong Yang*[1,2,3], Peiran Ren[1], Dongqing Zhang[1], Dong Chen[1], Fang Wen[1], Hongdong Li[2], Gang Hua[1]
[1]Microsoft Research [2]The Australian National University [3]Beijing Institute of Technology

Abstract

We present a Neural Aggregation Network (NAN) for video face recognition. The network takes a face video or face image set of a person with a variable number of face images as its input, and produces a compact and fixed-dimension feature representation. The whole network is composed of two modules. The feature embedding module is a deep Convolutional Neural Network (CNN), which maps each face image into a feature vector. The aggregation module consists of two attention blocks driven by a memory storing all the extracted features. It adaptively aggregates the features to form a single feature inside the convex hull spanned by them. Due to the attention mechanism, the aggregation is invariant to the image order. We found that NAN learns to advocate high-quality face images while repelling low-quality ones such as blurred, occluded and improperly exposed faces. The experiments on IJB-A, YouTube Face, Celebrity-1000 video face recognition benchmarks show that it consistently outperforms standard aggregation methods and achieves state-of-the-art accuracies.

1. Introduction

Video face recognition has caught more and more attention from the community in recent years [41, 22, 42, 11, 27, 23, 24, 29, 16, 37, 33, 10]. Compared to image-based face recognition, more information of the subjects can be exploited from input videos, which naturally incorporate faces of the same subject in varying poses and illumination conditions. The key issue in video face recognition is to build an appropriate visual representation of the video face, such that it can effectively integrate the information across different frames together, maintaining beneficial while discarding noisy information.

A naive approach would be representing a video face as a set of frame-level face features [37, 33]. Such a representation comprehensively maintains all the information across all frames. However, in the testing phase of face verification, to compare two video faces, one needs to fuse the matching results across all pairs of frames between the two face videos. Let n be the average number of video frames. The computational complexity is $O(n^2)$ per match operation, which is obviously not desirable. Besides, such a set-based representation would incur $O(n)$ space complexity per video face example, which demands a lot of memory storage and confronts efficient indexing.

We argue that it is more desirable to come with a compact, *fixed-size* visual representation at the video level, irrespective of the varied length of the video clips. Such a representation would allow direct, *constant-time* computation of the similarity or distance between two face videos without the need for frame-to-frame matching. A straightforward solution might be to extract a feature representation at each frame and then conduct a certain type of pooling to aggregate the frame-level features together to form a video level representation.

The most commonly adopted pooling strategies may be average and max pooling. There were some works successfully applying such pooling strategy in building video

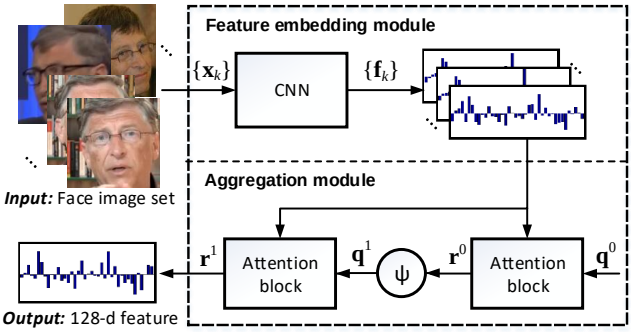

Figure 1. The face recognition framework of our method. All input faces images $\{\mathbf{x}_k\}$ are processed by a feature embedding module with a CNN, yielding a set of feature representations $\{\mathbf{f}_k\}$. These features are passed to the aggregation module, producing a 128-dimensional vector representation $\mathbf{r}^1$ for the input video faces. This compact representation is used for recognition.

*The main part of this work was done when J. Yang was an intern at MSR supervised by G. Hua.

face representations [30, 23, 7, 9]. For example, the Eigen-PEP representations [23] take the average of a part-based representation across different video frames and then conducts a PCA dimension reduction. Average pooling of face image features was also used in [30] and [7]. Both average and max pooling strategies were tested in the work of [9]. Other works such as the Video Fisher Vector Faces (VF^2) [29] have attempted to use a more general feature encoding schemes, *i.e.*, Fisher Vector coding, to aggregate local features across different video frames together to form a video-level representation.

Notwithstanding the demonstrated success of these methods, we believe that a good pooling or aggregation strategy should *adaptively* weigh and combine the frame-level features across all frames. The intuition is simple: a video (especially a long video sequence) or an image set may contain face images captured at various conditions of lighting, resolution, head pose *etc*., and a smart algorithm should favor face images that are more discriminative (or better "memorizable") and prevent poor face images from jeopardizing the recognition.

To this end, we look for an adaptive weighting scheme to linearly combine all frame-level features from a video together to form a compact and discriminative face representation. Different from the previous methods, we neither fix the weights nor rely on any particular heuristics to set them. Instead, we designed a neural network to adaptively calculate the weights. We named our network the Neural Aggregation Network (NAN), whose coefficients can be trained through supervised learning in a normal face recognition training task without the need for extra supervision signals.

The proposed NAN is composed of two major modules that could be trained end-to-end or one by one separately. The first one is a feature embedding module which serves as a frame-level feature extractor using a deep CNN model. The other is the aggregation module that adaptively fuses the feature vectors of all the video frames together.

Our neural aggregation network is designed to inherit the main advantages of pooling techniques, including the ability to handle arbitrary input size and producing order-invariant representations. The key component of this network is inspired by the Neural Turing Machine [13] and the Orderless Set Network [38], both of which applied an attention mechanism to organize the input through a memory. This mechanism can take an input of arbitrary size and work as a tailor emphasizing or suppressing each input element just as in a weighted averaging, and very importantly it is order independent. Instead of using a Long Short Term Memory network (LSTM) [15] as in the work of [38], we found in our experiments that a simple network structure with two cascaded attention blocks performs well for the face recognition task.

Apart from building a video-level representation, the neural aggregation network can also serve as a subject level feature extractor to fuse multiple data sources. For example, one can feed it with all available images and videos, or the aggregated video-level features of multiple videos from the same subject, to obtain a single feature representation with fixed size. In this way, the face recognition system not only enjoys the time and memory efficiency due to the compact representation, but also exhibits superior performance, as we will show in our experiments.

We evaluated the proposed NAN for both the tasks of video face verification and identification. We observed consistent margins in three challenging datasets, including the YouTube Face dataset [41], the IJB-A dataset [21], and the Celebrity-1000 dataset [24], compared to the baseline strategies and other competing methods. In summary, this work contributes to the following aspects:

- We proposed an end-to-end trainable neural aggregation network, which learns an adaptive, content-aware pooling strategy for a set of visual features.
- We applied it for the task of video face recognition, which leads to a comprehensive, compact, and yet discriminative video face representation.
- We demonstrated its superiority over various widely-adopted strategies, and achieved state-of-the-art recognition accuracy on challenging video face recognition datasets.

Last but not least, we shall point out that our proposed NAN can serve as a general framework for learning content-adaptive pooling. Therefore, it may also serve as a feature aggregation scheme for other computer vision tasks such as image recognition and video event recognition [19, 12, 20, 2]. We leave this as our future work to evaluate its performance against other pooling schemes on these tasks, such as VLAD [19, 12, 20] and scene aligned pooling [2].

2. Neural Aggregation Network

As shown in Fig. 1, the NAN network takes a set of face images of a person as input and outputs a single feature vector as its representation for the recognition task. It stands on the shoulders of a modern deep CNN model, and becomes more powerful for video face recognition by adaptively aggregating all frames in the video into a compact vector representation. It is composed of two modules: a feature embedding module and an aggregation module.

In this section, we will first introduce the feature embedding module and the aggregation module in Section 2.1 and 2.2, respectively, then present the training strategies in Section 2.3.

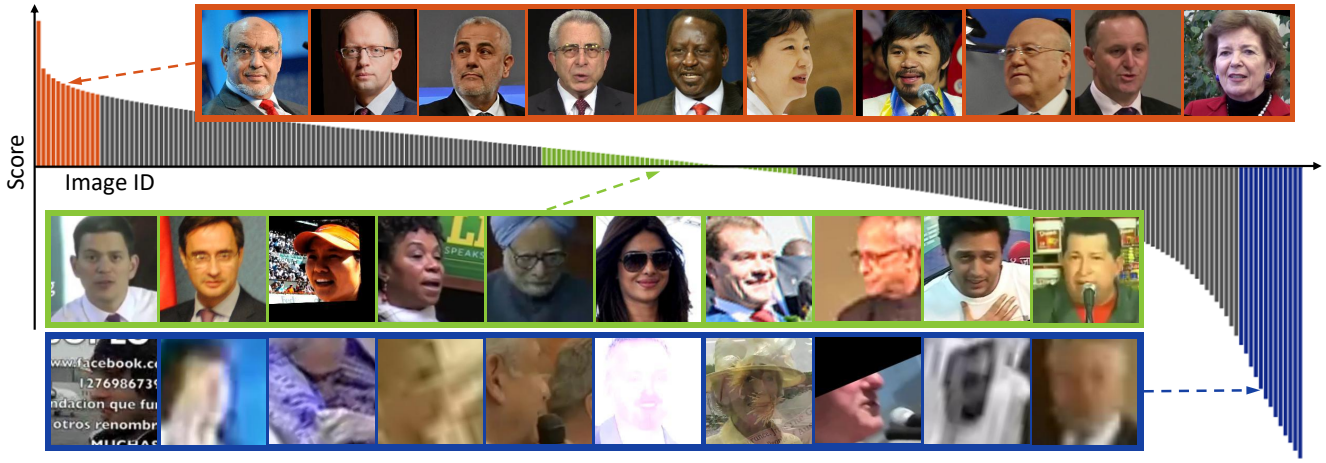

Figure 2. Face images in the IJB-A dataset, sorted by their scores (values of e in Eq. 2) from a single attention block trained in the face recognition task. The faces in the top, middle and bottom rows are sampled from the faces with scores in the highest 5%, a 10% window centered at the median, and the lowest 5%, respectively.

2.1. Feature embedding module

The image embedding module of our NAN is a deep Convolution Neural Network (CNN), which embeds each frame of a video to a face feature representation. To leverage modern deep CNN networks with high-end performances, in this paper we adopt the GoogLeNet [36] with the Batch Normalization (BN) technique [18]. Certainly, other network architectures are equally applicable here as well. The GoogLeNet produces 128-dimension image features, which are first *normalized* to be unit vectors then fed into the aggregation module. In the rest of this paper, we will simply refer to the employed GoogLeNet-BN network as CNN.

2.2. Aggregation module

Consider the video face recognition task on n pairs of video face data $(\mathcal{X}^i, y_i)_{i=1}^n$, where $\mathcal{X}^i$ is a face video sequence or a image set with varying image number K_i, *i.e.* $\mathcal{X}^i = \{\mathbf{x}_1^i, \mathbf{x}_2^i, ..., \mathbf{x}_{K_i}^i\}$ in which $\mathbf{x}_k^i, k = 1, ..., K_i$ is the k-th frame in the video, and y_i is the corresponding subject ID of $\mathcal{X}^i$. Each frame $\mathbf{x}_k^i$ has a corresponding normalized feature representation $\mathbf{f}_k^i$ extracted from the feature embedding module. For better readability, we omit the upper index where appropriate in the remaining text. Our goal is to utilize all feature vectors from a video to generate a set of linear weights $\{a_k\}_{k=1}^t$, so that the aggregated feature representation becomes

$$\mathbf{r} = \sum_k a_k \mathbf{f}_k. \tag{1}$$

In this way, the aggregated feature vector has the same size as a single face image feature extracted by the CNN.

Obviously, the key of Eq. 1 is its weights $\{a_k\}$. If $a_k \equiv \frac{1}{t}$, Eq. 1 will degrades to naive averaging, which is usually non-optimal, as we will show in our experiments. We instead try to design a better weighting scheme.

Three main principles have been considered in designing our aggregation module. First, the module should be able to process different numbers of images (*i.e.* different K_i's), as the video data source varies from person to person. Second, the aggregation should be invariant to the image order – we prefer the result unchanged when the image sequence are reversed or reshuffled. As such, the aggregation module can handle an arbitrary set of image or video faces without temporal information (*e.g.* that collected from different Internet locations). Third, the module should be adaptive to the input faces and has parameters trainable through supervised learning in a standard face recognition training task.

We found the *memory attention mechanism* described in [13, 34, 38] suits well the above goals. The idea therein is to use a neural model to read external memories through a differentiable addressing/attention scheme. Such models are often coupled with Recurrent Neural Networks (RNN) to handle sequential inputs/outputs [13, 34, 38]. However, the core idea, *i.e.* the attention mechanism is applicable to our aggregation task. In this work, we treat the face features as the memory, and cast feature weighting as a memory addressing procedure. We employ in the aggregation module the "attention blocks", to be described as follows.

2.2.1 Attention blocks

An attention block reads all feature vectors from the feature embedding module, and generate linear weights for them. Specifically, let $\{\mathbf{f}_k\}$ be the face feature vectors, then an attention block filters them with a kernel $\mathbf{q}$ via dot product, yielding a set of corresponding significances $\{e_k\}$. They are then passed to a softmax operator to generate positive

Table 1. Performance comparison on the IJB-A dataset. TAR/FAR: True/False Accept Rate for verification. TPIR/FPIR: True/False Positive Identification Rate for identification.

Method	1:1 Verification TAR@FAR of: 0.001	0.01	1:N Identification TPIR@FPIR of: 0.01	0.1
CNN+AvgPool	0.771	0.913	0.634	0.879
NAN single attention	0.847	0.927	0.778	0.902
NAN cascaded attention	0.860	0.933	0.804	0.909

weights $\{a_k\}$ with $\sum_k a_k = 1$. These two operations can be described by the following equations, respectively:

$$e_k = \mathbf{q}^T \mathbf{f}_k \tag{2}$$

$$a_k = \frac{\exp(e_k)}{\sum_j \exp(e_j)}. \tag{3}$$

Therefore, our algorithm essentially selects one point inside of the convex hull spanned by all the feature vectors. One related work is [3] where each face image set is approximated with a convex hull and set similarities are defined as the shortest path between two convex hulls.

In this way, the number of inputs $\{\mathbf{f}_k\}$ does not affect the size of aggregation $\mathbf{r}$, which is of the same dimension as a single feature $\mathbf{f}_k$. Besides, the aggregation result is invariant to the input order of $\mathbf{f}_k$: according to Eq. 1, 2, and 3 and, permuting $\mathbf{f}_k$ and $\mathbf{f}_{k'}$ has no effects on the aggregated representation $\mathbf{r}$. Furthermore, an attention block is modulated by the filter kernel $\mathbf{q}$, which can be trained as described as follows.

Single attention block – Universal face feature quality measurement. We first tried using one attention block for aggregation. In this case, vector $\mathbf{q}$ is the parameter to learn. It has the same size as a single feature $\mathbf{f}$ and serves as a universal prior measuring the face feature quality.

We trained the network to perform video face verification (see Section 2.3 for details) in the IJB-A dataset [21] on the extracted face features, and Fig. 2 shows the sorted scores of all the faces images in the dataset. It can be seen that after training, the network favors high-quality face images, such as those of high resolutions and relatively simple backgrounds. It down-weights face images with blur, occlusion, improper exposure and extreme poses. Table 1 shows that the network achieves higher accuracy the average pooling baseline on the dataset in the verification and identification tasks of the dataset.

Cascaded two attention blocks – Content-aware aggregation. We believe a content-aware aggregation can perform even better. The intuition behind is that face image variation may be expressed differently at different geographic locations in the feature space (*i.e.* for different persons), and content-aware aggregation can learn to select features that are more discriminative for the identity of the input image set. To this end, we employed two attention blocks in a cascaded and end-to-end fashion.

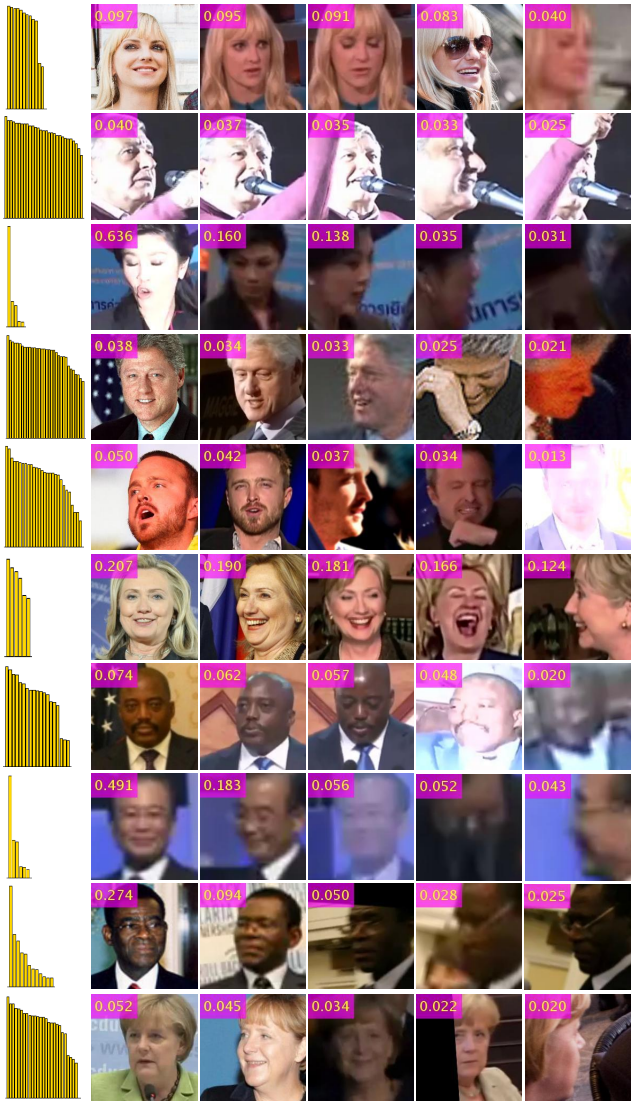

Figure 3. Typical examples showing the weights of the images in the image sets computed by the NAN. In each row, the leftmost bar chart shows the sorted weights of all the images in the set, and the five sampled face images are sorted according to their weights displayed in the top left corner. The top and bottom five rows are from the training and testing data, respectively.

Let $\mathbf{q}^0$ be the kernel of the first attention block, and $\mathbf{r}^0$ be the aggregated feature with $\mathbf{q}^0$. We adaptively compute $\mathbf{q}^1$, the kernel of the second attention block, through a transfer layer taking $\mathbf{r}^0$ as the input:

$$\mathbf{q}^1 = \tanh(\mathbf{W}\mathbf{r}^0 + \mathbf{b}) \tag{4}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the weight matrix and bias vector of the

neurons respectively, and $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ imposes the hyperbolic tangent nonlinearity. The feature vector $\mathbf{r}^1$ generated by $\mathbf{q}^1$ will be the final aggregation results. Therefore, $(\mathbf{q}^0, \mathbf{W}, \mathbf{b})$ are now the trainable parameters of the aggregation module.

We trained the network on the IJB-A dataset again, and Table 1 shows that the network obtained better results than using single attention block. Figure 3 shows some typical examples of the weights computed by the trained network for different image sets

More attention blocks. We also tried more than two attention blocks cascaded for aggregation, similar to a recurrent network structure as in [13, 34, 38]. However, we did not observe better results, despite more computation were brought in. We are convinced that cascading two attention blocks is a better option and hence adopt such a structure (as in Fig. 1) in the experiment section.

2.3. Network training

The NAN network is trained either for both face verification and identification tasks with standard configurations.

For verification, we build a *siamese* neural aggregation network structure [8] with two NANs sharing weights, and minimize the average contrastive loss [14]: $l_{i,j} = y_{i,j}||\mathbf{r}_i^1 - \mathbf{r}_j^1||_2^2 + (1 - y_{i,j}) \max(0, m - ||\mathbf{r}_i^1 - \mathbf{r}_j^1||_2^2)$, where $y_{i,j} = 1$ if the pair (i, j) is from the same identity and $y_{i,j} = 0$ otherwise. The constant m is set to 2 in all our experiments.

For identification, we add on top of NAN a fully-connected layer and minimize the average classification loss: $l_i = -\log p_{i,y_i}$ where y_i is the target label of the i-th video instance, $p_{i,y_i} = \frac{\exp(p_{i,y_i})}{\sum_z \exp(p_{i,z})}$, and $p_{i,z}$ is the zth output of the FC layer.

2.3.1 Module training

The two modules can either be trained simultaneously in an end-to-end fashion, or separately one by one. In this paper, we take the latter option to train the NAN network. Specifically, we first train the CNN on single images with the identification task, then we train the aggregation module on top of the features extracted by CNN.

We chose this separate training strategy mainly for two reasons. First, in this work we would like to focus on analyzing the effectiveness and performance of the aggregation module with the attention mechanism. Despite the huge success of applying deep CNN in image-based face recognition task, little attention has been drawn to CNN feature aggregation to our knowledge. Second, training a deep CNN usually necessitates a large volume of labeled data. While millions of still images can be obtained for training nowadays [37, 33], it appears not practical to collect such amount of distinctive face videos or sets. We leave an end-to-end training of the NAN as our future work.

3. Experiments

This section evaluates the performance of the proposed NAN network. We will begin with introducing our training details and the baseline methods, followed by reporting the results on three video face recognition datasets: the IARPA Janus Benchmark A (IJB-A) [21], the YouTube Face dataset [41], and the Celebrity-1000 dataset [24]. Due to space limitation, more results are presented in the supplementary material.

3.1. Training details

As mentioned in Section 2.3, two networks are trained separately in this paper. To train the CNN, we use around 3M face images of 50K identities crawled from the internet to perform image-based identification. The faces are detected using the JDA method [5], and aligned with the LBF method [31]. The input image size is 224x224. After training, the CNN is fixed and we focus on analyzing the effectiveness of the neural aggregation module.

The aggregation module is trained on each video face dataset we tested on with standard backpropagation and gradient descent. As the network is quite simple and image features are compact (128-d), the training process is quite efficient: training on 5K video pairs with $\sim$1M images in total only takes less than 2 minutes on a CPU of a desktop PC.

3.2. Baseline methods

The performance of the NAN is evaluated against some baseline methods. Since our goal is to build a compact representation for video faces, we compared its results with simple aggregation strategies such as average pooling. We also compared it with some set-to-set similarity measurements leveraging pairwise comparison on the image level. To keep it simple, *we primarily use the L_2 feature distances for face recognition* (all features are normalized), although it is possible to combine with an extra metric learning or template adaption technique [10] to further boost the performance on each dataset.

In the baseline methods, *CNN+Min* L_2, *CNN+Max* L_2, *CNN+Mean* L_2 and *CNN+SoftMin* L_2 measure the similarity of two video faces based on the L_2 feature distances of all frame pairs. These methods necessitate storing all image features of a video, *i.e.* with $O(n)$ space complexity. The first three use respectively the minimum, maximum and mean pairwise distance, thus having $O(n^2)$ complexity for similarity computation. *CNN+SoftMin* L_2 corresponds to the SoftMax similarity score advocated in some works such

Table 2. Performance evaluation on the IJB-A dataset. For verification, the true accept rates (TAR) *vs.* false positive rates (FAR) are reported. For identification, the true positive identification rate (TPIR) *vs.* false positive identification rate (TPIR) and the Rank-N accuracies are presented. (†: first aggregating the images in each media then aggregate the media features in a template. *: results cited from [10].)

Method	1:1 Verification TAR			1:N Identification TPIR				
	FAR=0.001	FAR=0.01	FAR=0.1	FPIR=0.01	FPIR=0.1	Rank-1	Rank-5	Rank-10
B-CNN [9]	–	–	–	0.143 ± 0.027	0.341 ± 0.032	0.588 ± 0.020	0.796 ± 0.017	–
LSFS [39]	0.514 ± 0.060	0.733 ± 0.034	0.895 ± 0.013	0.383 ± 0.063	0.613 ± 0.032	0.820 ± 0.024	0.929 ± 0.013	–
DCNN$_{manual}$+metric[7]	–	0.787 ± 0.043	0.947 ± 0.011	–	–	0.852 ± 0.018	0.937 ± 0.010	0.954 ± 0.007
Triplet Similarity [32]	0.590 ± 0.050	0.790 ± 0.030	0.945 ± 0.002	$0.556 \pm 0.065^*$	$0.754 \pm 0.014^*$	$0.880 \pm 0.015^*$	0.95 ± 0.007	$0.974 \pm 0.005^*$
Pose-Aware Models [25]	0.652 ± 0.037	0.826 ± 0.018	–	–	–	0.840 ± 0.012	0.925 ± 0.008	0.946 ± 0.007
Deep Milti-Pose [1]	–	0.876	0.954	0.52^*	0.75^*	0.846	0.927	0.947
DCNN$_{fusion}$ [6]	–	0.838 ± 0.042	0.967 ± 0.009	$0.577 \pm 0.094^*$	$0.790 \pm 0.033^*$	0.903 ± 0.012	0.965 ± 0.008	0.977 ± 0.007
Masi *et al*. [26]	0.725	0.886	–	–	–	0.906	0.962	0.977
Triplet Embedding [32]	0.813 ± 0.02	0.90 ± 0.01	0.964 ± 0.005	0.753 ± 0.03	0.863 ± 0.014	0.932 ± 0.01	–	0.977 ± 0.005
VGG-Face [30]	–	$0.805 \pm 0.030^*$	–	$0.461 \pm 0.077^*$	$0.670 \pm 0.031^*$	$0.913 \pm 0.011^*$	–	$0.981 \pm 0.005^*$
Template Adaptation[10]	0.836 ± 0.027	0.939 ± 0.013	$\mathbf{0.979 \pm 0.004}$	0.774 ± 0.049	0.882 ± 0.016	0.928 ± 0.010	0.977 ± 0.004	$\mathbf{0.986 \pm 0.003}$
CNN+Max L_2	0.202 ± 0.029	0.345 ± 0.025	0.601 ± 0.024	0.149 ± 0.033	0.258 ± 0.026	0.429 ± 0.026	0.632 ± 0.033	0.722 ± 0.030
CNN+Min L_2	0.038 ± 0.008	0.144 ± 0.073	0.972 ± 0.006	0.026 ± 0.009	0.293 ± 0.175	0.853 ± 0.012	0.903 ± 0.010	0.924 ± 0.009
CNN+Mean L_2	0.688 ± 0.080	0.895 ± 0.016	0.978 ± 0.004	0.514 ± 0.116	0.821 ± 0.040	0.916 ± 0.012	0.973 ± 0.005	0.980 ± 0.004
CNN+SoftMin L_2	0.697 ± 0.085	0.904 ± 0.015	0.978 ± 0.004	0.500 ± 0.134	0.831 ± 0.039	0.919 ± 0.010	0.973 ± 0.005	0.981 ± 0.004
CNN+MaxPool	0.202 ± 0.029	0.345 ± 0.025	0.601 ± 0.024	0.079 ± 0.005	0.179 ± 0.020	0.757 ± 0.025	0.911 ± 0.013	0.945 ± 0.009
CNN+AvePool	0.771 ± 0.064	0.913 ± 0.014	0.977 ± 0.004	0.634 ± 0.109	0.879 ± 0.023	0.931 ± 0.011	0.972 ± 0.005	0.979 ± 0.004
CNN+AvePool†	0.856 ± 0.021	0.935 ± 0.010	0.978 ± 0.004	0.793 ± 0.044	0.909 ± 0.011	0.951 ± 0.005	0.976 ± 0.004	0.984 ± 0.004
NAN	0.860 ± 0.012	0.933 ± 0.009	$\mathbf{0.979 \pm 0.004}$	0.804 ± 0.036	0.909 ± 0.013	0.954 ± 0.007	0.978 ± 0.004	0.984 ± 0.003
NAN†	$\mathbf{0.881 \pm 0.011}$	$\mathbf{0.941 \pm 0.008}$	0.978 ± 0.003	$\mathbf{0.817 \pm 0.041}$	$\mathbf{0.917 \pm 0.009}$	$\mathbf{0.958 \pm 0.005}$	$\mathbf{0.980 \pm 0.005}$	$\mathbf{0.986 \pm 0.003}$

as [25, 26, 1]. It has $O(m*n^2)$ complexity for computation[1].

CNN+MaxPool and *CNN+AvePool* are respectively max-pooling and average-pooling along each feature dimension for aggregation. These two methods as well as our NAN produce a 128-d feature representation for each video and compute the similarity in $O(1)$ time.

3.3. Results on IJB-A dataset

The IJB-A dataset contains face images and videos captured from unconstrained environments. There are 500 subjects with 5,397 images and 2,042 videos sampled to 20,412 frames in total, 11.4 images and 4.2 videos per subject on average. The IJB-A dataset features full pose variation and wide variations in imaging conditions, which makes the face recognition very challenging. We detect the faces with landmarks using STN [4] face detector, and then align the face image with similarity transformation.

In this dataset, each training and testing instance is called a 'template' and comprises a mixture of still images and sampled video frames. Each still image or a set of video frames from the same source is called a media. The numbers of images in the templates range from 1 to 190 with approximately 10 images per template on average. There are 10 training and testing splits. Each of them contains 333 subjects, and its corresponding testing split takes the other 167 subjects.

Since one template may contain multiple media and the dataset provides the media id for each image, another possible aggregation strategy is first aggregating the frames in each media then aggregate the media features again to generate the template feature. This strategy is used in [10, 32] and also tested in this work with *CNN+AvePool* and our NAN. Note that the media id information may not be always available in practical systems.

We tested the proposed method on both the 'compare' protocol for *1:1 face verification* and the 'search' protocol for *1:N face identification*. For verification, the true accept rates (TAR) vs. false positive rates (FAR) are reported. For identification, the true positive identification rate (TPIR) vs. false positive identification rate (TPIR) and the Rank-N accuracies are reported. Table 2 presents the numerical results of different methods, and Figure 4 shows the receiver operating characteristics (ROC) curves for verification as well as the cumulative match characteristic(CMC) and decision error trade off (DET) curves for identification. The metrics are calculated according to [21, 28] on 10 splits.

In general, *CNN+Max* L_2, *CNN+Min* L_2 and *CNN+MaxPool* performed worst among the baseline methods. *CNN+SoftMin* L_2 performed slightly better than *CNN+MaxPool*. The use of media id significantly improved the performance of *CNN+AvePool*, but gave a relatively small help for NAN. This suggests that NAN is more robust to handle tough templates possibly dominated

[1] m is the number of scaling factor β used (see [26] for details). We tested 20 combinations of (negative) β's, including single [1] or multiple values [25, 26], and report the best results obtained.

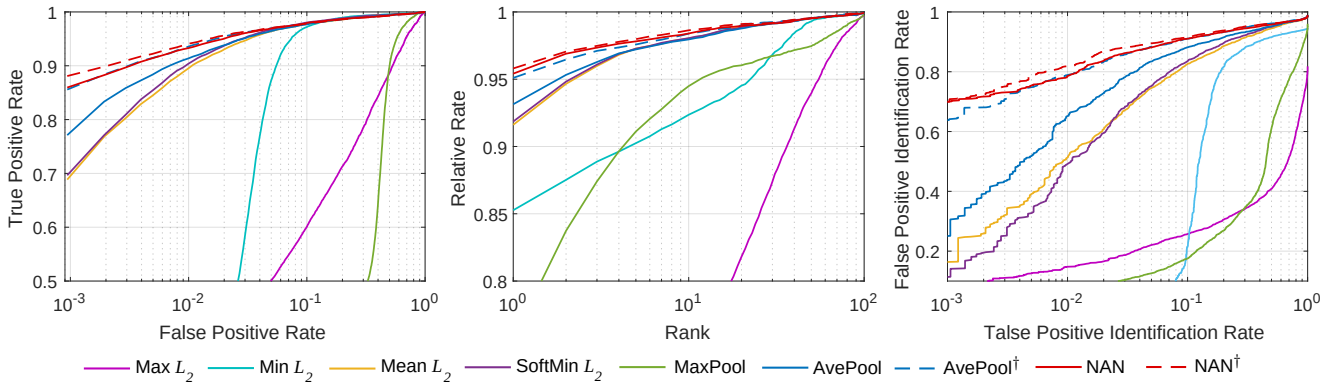

Figure 4. Average ROC (Left), CMC (Middle) and DET (Right) curves of the NAN and the baselines on the IJB-A dataset over 10 splits.

Table 3. Verification accuracy comparison of state-of-the-art methods, our baselines and NAN network on the YouTube Face dataset.

Method	Accuracy (%)	AUC
LM3L [17]	81.3 ± 1.2	89.3
DDML(combined)[16]	82.3 ± 1.5	90.1
EigenPEP [23]	84.8 ± 1.4	92.6
DeepFace-single [37]	91.4 ± 1.1	96.3
DeepID2+ [35]	93.2 ± 0.2	–
Wen *et al*. [40]	94.9	–
FaceNet [33]	95.12 ± 0.39	–
VGG-Face [30]	97.3	–
CNN+Max. L_2	91.96 ± 1.1	97.4
CNN+Min. L_2	94.96 ± 0.79	98.5
CNN+Mean L_2	95.30 ± 0.74	98.7
CNN+SoftMin L_2	95.36 ± 0.77	98.7
CNN+MaxPool	88.36 ± 1.4	95.0
CNN+AvePool	95.20 ± 0.76	98.7
NAN	$\mathbf{95.72 \pm 0.64}$	**98.8**

by poor images in a few media. Without the media aggregation, NAN outperforms all its baselines by appreciable margins, especially on the low FAR cases. For example, in the verification task, the TARs of our NAN at FARs of 0.001 and 0.01 are respectively 0.860 and 0.933, reducing the errors of the best results from its baselines by about 39% and 23%, respectively.

To our knowledge, the NAN achieves top performances compared with the previous methods with the media aggregation. It has a same verification TAR at FAR=0.1 and identification Rank-10 CMC as the method of [10], but outperforms it on all other metrics (*e.g*. 0.881 *vs*. 0.838 TARs at FAR=0.01, 0.817 *vs*. 0.774 TPIRs at FPIR=0.01 and 0.958 *vs*. 0.928 Rank-1 accuracy).

Figure 3 has shown some typical examples of the weighting results by the the NAN. It exhibits the ability to choose high-quality and more discriminative face images, while repelling poor face images.

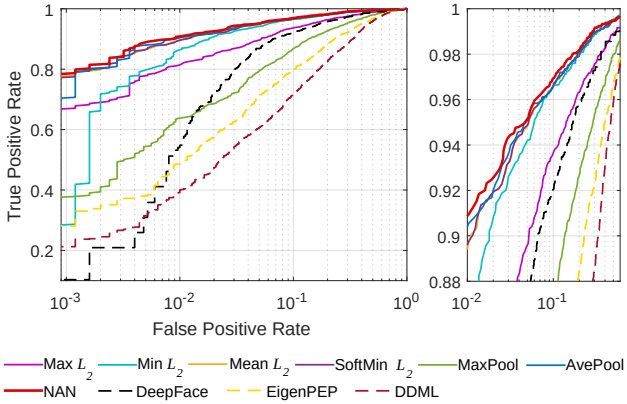

Figure 5. Average ROC curves of different baseline methods and the proposed NAN method on the YTF dataset over 10 splits.

3.4. Results on YouTube Face dataset

We then tested our method on the YouTube Face (YTF) dataset which is designed for unconstrained *face verification* in videos. It contains 3,425 videos of 1,595 different people, and an average of 2.15 videos are available for each subject. The video clip lengths vary from 48 to 6,070 frames, with an average length 181.3 frames per video. Ten folds of 500 video pairs are available, and we follow the standard verification protocol to report the average accuracy with cross-validation. We again use the STN and similarity transformation to align the face images.

The results of our NAN, its baselines, and other methods are presented in Table 3, with their ROC curves shown in Fig. 5. It can be seen that the NAN again outperforms all its baselines. However, the gaps between NAN and the best-performing baselines are smaller compared to the results on the IJB-A dataset. This is because the face variations in this dataset are relatively small (compare the examples in Fig. 6 and Fig. 3), thus no much beneficial information can be extracted compared to naive average pooling or computing mean L_2 distances.

Compared to previous methods, the NAN achieves a

Figure 6. Typical examples on the YTF dataset showing the weights of the images in the image sets computed by the NAN. The leftmost bar chart shows the sorted weights of all the images in the set, and the five sampled face images are sorted according to their weights displayed in the top left corner. The top and bottom three rows are from the training and testing data, respectively.

mean accuracy of **95.72%**, reducing the error of FaceNet by 12.3%. Note that FaceNet is also based on a GoogLeNet style network, and the average similarity of all pairs of 100 frames in each video (*i.e.*, 10K pairs) was used [33]. To our knowledge, only the VGG-Face [30] achieves an accuracy (97.3%) higher than ours. However, that result is based on a further discriminative metric learning on YTF, without which the accuracy is only 91.5%.

3.5. Results on Celebrity-1000 dataset

The Celebrity-1000 dataset is designed to study the unconstrained video-based *face identification* problem. It contains 159,726 video sequences of 1,000 human subjects, with 2.4M frames in total (∼15 frames per sequence). We use the provided 5 facial landmarks to align the face images.

Two types of protocols – open-set and close-set – exist on this dataset. In the open-set protocol, 200 subjects are used for training, while video sequences of the rest 800 subjects are used as the gallery set and probe set at the testing stage. There are 4 different settings with different numbers of probe and gallery subjects: 100, 200, 400 and 800. In the close-set protocol, the video sequences from all 1,000 subjects are divided into a training (gallery) subset and a testing (probe) subset. There are also 4 settings for close-set: 100, 200, 500 and 1000 subjects.

Table 4. Identification performance (rank-1 accuracies, %) on the Celebrity-1000 dataset with the *close-set* protocol.

Method	Number of Subjects			
	100	200	500	1000
MTJSR [24]	50.60	40.80	35.46	30.04
Eigen-PEP [23]	50.60	45.02	39.97	31.94
CNN+AvePool - VideoAggr	86.06	82.38	80.48	74.26
CNN+AvePool - SubjectAggr	84.46	78.93	77.68	73.41
NAN - VideoAggr	88.04	82.95	**82.27**	76.24
NAN - SubjectAggr	**90.44**	**83.33**	**82.27**	**77.17**

Close-set tests. For the close-set protocol, we first trained the network on the video sequences by minimizing the identification loss. The FC layer output values are taken as the scores, and the subject with the maximum score is the identification result. We also trained a linear classifier for our baseline method *CNN+AvePool* to classify each video feature. As the features are built on video sequences, we call this approach 'VideoAggr' to distinguish it from another approach to be described next. Each subject in the dataset has multiple video sequences, thus we can naturally build a single representation for a subject by aggregating all available images in all the training (gallery) video sequences. We call this approach 'SubjectAggr'. In this way, the linear classifier can be bypassed, and identification can be achieved simply by comparing the feature L_2 distances.

The results are presented in Table 4. All the baseline methods and NAN outperformed previous methods by large margins, and the NAN consistently outperformed the baseline methods on all these tasks. For the 'VideoAggr' approach, NAN reduces the errors of *CNN+AvePool* by 14%, 9%, 8% and 3% for the settings of 100, 200, 500, and 1000 subjects, respectively. Figure 7 shows some typical results of the video sequences and weights computed by the NAN.

For the 'SubjectAggr' approach, it brings significant improvements upon the baseline: the errors are reduced by 38%, 21%, 21% and 14% respectively. It is interesting to see that, 'SubjectAggr' leads to a clear performance drop by *CNN+AvePool*. This indicates that the hand-crafted aggregation gets even worse when applied on the subject level with multiple videos. However, our NAN can benefit from 'SubjectAggr', yielding results consistently better than or on par with the 'VideoAggr' approach and delivers a considerable accuracy boost (*e.g.* for 100 subjects the error is reduced by 20%). This indicates that our NAN works very well on handling large data variations.

Open-set tests. We then tested our NAN with the *close-set* protocol. We first trained the network on the provided training video sequences. In the testing stage, we took the 'SubjectAggr' approach described before to build a highly-compact face representation for each gallery subject. Identification was performed similarly by comparing the L_2 dis-

Figure 7. Typical examples on the Celebrity-1000 dataset (close-set tests) showing the weights of the images in the image sets computed by the NAN. The leftmost bar chart shows the sorted weights of all the images in the set, and the five sampled face images are sorted according to their weights displayed in the top left corner. The top and bottom three rows are from the training and testing data, respectively.

Table 5. Identification performance (rank-1 accuracies, %) on the Celebrity-1000 dataset with the *open-set* protocol.

	Number of Subjects			
Method	100	200	400	800
MTJSR [24]	46.12	39.84	37.51	33.50
Eigen-PEP [23]	51.55	46.15	42.33	35.90
CNN+Mean L_2	84.88	79.88	76.76	70.67
CNN+AvePool - SubjectAggr	84.11	79.09	78.40	75.12
NAN - SubjectAggr	**88.76**	**85.21**	**82.74**	**79.87**

tances between aggregated face representations.

The results in Table 5 shows that our NAN reduce the error of the baseline *CNN+AvePool* by 29%, 29%, 20%, 19% for the settings of 100, 200, 400, and 800 subjects, respectively. This again suggests that in the presence of large face variances, the widely used strategies such as average-pooling aggregation and the pairwise distance computation are far from optimal. In such cases, the learned NAN model is powerful and can yield much superior results. The aggravated feature representation produced by NAN is more favorable for the video face recognition task.

4. Conclusions

We have presented a Neural Aggregation Network for video face representation and recognition. It fuses all input frames with a set of content adaptive weights, resulting in a compact representation that is invariant to the input frame order. The aggregation scheme is simple with small computation and memory footprints, but can generate a quality face representation after trained through supervised learning. The experiments have shown that the proposed NAN network consistently outperforms the baselines aggregation strategies and achieves top performances on the public datasets.

It should be noted that the proposed method can support general set representation, and therefore can be used in applications other than video face recognition.

References

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

[2] L. Cao, Y. Mu, A. Natsev, S.-F. Chang, G. Hua, and J. R. Smith. Scene aligned pooling for complex video recognition. In *European Conference on Computer Vision (ECCV)*, pages 688–701. 2012.

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

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

[5] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *European Conference on Computer Vision (ECCV)*, pages 109–122. 2014.

[6] J.-C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep cnn features. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.

[7] J.-C. Chen, R. Ranjan, A. Kumar, C.-H. Chen, V. Patel, and R. Chellappa. An end-to-end system for unconstrained face verification with deep convolutional neural networks. In *IEEE International Conference on Computer Vision Workshops*, pages 118–126, 2015.

[8] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 539–546, 2005.

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

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

[11] Z. Cui, W. Li, D. Xu, S. Shan, and X. Chen. Fusing robust face region descriptors via multiple metric learning for face recognition in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3554–3561, 2013.

[12] M. Douze, J. Revaud, C. Schmid, and H. Jégou. Stable hyper-pooling and query expansion for event detection. In *International Conference on Computer Vision (ICCV)*, pages 1825–1832, 2013.

[13] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *CoRR*, abs/1410.5401, 2014.

[14] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1735–1742, 2006.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

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

[17] J. Hu, J. Lu, J. Yuan, and Y.-P. Tan. Large margin multi-metric learning for face and kinship verification in the wild. In *Asian Conference on Computer Vision (ACCV)*, pages 252–267. 2014.

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

[19] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3304–3311, 2010.

[20] H. Jégou and A. Zisserman. Triangulation embedding and democratic aggregation for image search. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3310–3317, 2014.

[21] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1931–1939, 2015.

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

[23] H. Li, G. Hua, X. Shen, Z. Lin, and J. Brandt. Eigen-PEP for video face recognition. In *Asian Conference on Computer Vision (ACCV)*, pages 17–33. 2014.

[24] L. Liu, L. Zhang, H. Liu, and S. Yan. Toward large-population face identification in unconstrained videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(11):1874–1884, 2014.

[25] I. Masi, S. Rawls, G. Medioni, and P. Natarajan. Pose-aware face recognition in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4838–4846, 2016.

[26] I. Masi, A. T. a. Trãn, J. Toy Leksut, T. Hassner, and G. Medioni. Do we really need to collect millions of faces for effective face recognition? In *European Conference on Computer Vision (ECCV)*, 2016.

[27] H. Mendez-Vazquez, Y. Martinez-Diaz, and Z. Chai. Volume structured ordinal features with background similarity measure for video face recognition. In *International Conference on Biometrics (ICB)*, 2013.

[28] M. Ngan and P. Grother. Face recognition vendor test (FRVT) performance of automated gender classification algorithms. In *Technical Report NIST IR 8052*. National Institute of Standards and Technology, 2015.

[29] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman. A compact and discriminative face track descriptor. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1693–1700, 2014.

[30] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference (BMVC)*, volume 1, page 6, 2015.

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

[32] S. Sankaranarayanan, A. Alavi, C. Castillo, and R. Chellappa. Triplet probabilistic embedding for face verification and clustering. *arXiv preprint arXiv:1604.05417*, 2016.

[33] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[34] S. Sukhbaatar, J. Weston, R. Fergus, et al. End-to-end memory networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2440–2448, 2015.

[35] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2892–2900, 2015.

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

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

[38] O. Vinyals, S. Bengio, and M. Kudlur. Order matters: sequence to sequence for sets. In *International Conference on Learning Representation*, 2016.

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

[40] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision (ECCV)*, pages 499–515, 2016.

[41] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 529–534, 2011.

[42] L. Wolf and N. Levy. The SVM-Minus similarity score for video face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3523–3530, 2013.