Neural Aggregation Network for Video Face Recognition

Jiaolong Yang[*1,2,3], Peiran Ren[1], Dong Chen[1], Fang Wen[1], Hongdong Li[2], Gang Hua[1]

[1]Microsoft Research, Beijing, China
[2]Research School of Engineering, Australian National University
[3]Beijing Lab of Intelligent Information Technology, Beijing Institute of Technology

Abstract

In this paper, 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 variable number of face frames as its input, and produces a compact and fixed-dimension visual representation of that person. The whole network is composed of two modules. The feature embedding module is a CNN which maps each face frame into a feature representation. The neural aggregation module is composed of two content based attention blocks which is driven by a memory storing all the features extracted from the face video through the feature embedding module. The output of the first attention block adapts the second, whose output is adopted as the aggregated representation of the video faces. Due to the attention mechanism, this representation is invariant to the order of the face frames. The experiments show that the proposed NAN consistently outperforms hand-crafted aggregations such as average pooling, and achieves state-of-the-art accuracies on three video face recognition datasets: the YouTube Face, IJB-A and Celebrity-1000 datasets.

1. Introduction

Video face recognition has caught more and more attention from the community in recent years [29, 17, 30, 5, 20, 18, 19, 21, 10, 26, 23]. 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
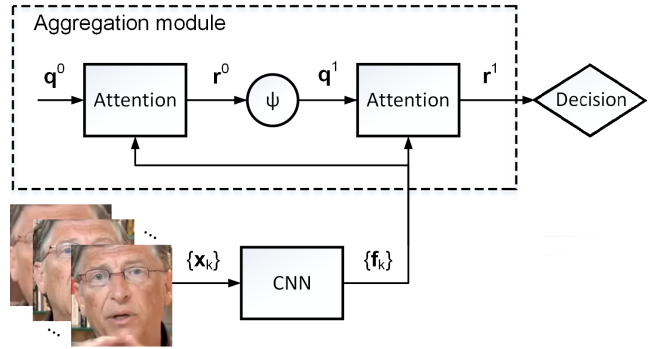
Figure 1. The face recognition framework of our method. All input faces $\{\mathbf{x}_k\}$ are processed by a feature embedding module with a CNN, yielding a set of feature representations, $\{\mathbf{f}_k\}$. These features are passed to the aggregation module, producing a 128-dimensional representation $\mathbf{r}^1$ for the input video faces. This compact representation can then be used for the decision.

noisy information.

A naive approach would be representing a video face as a set of frame-level face features [26, 23]. Such a representation comprehensively maintains all the information across all frames. However, in the recognition phase, 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 clip level for video faces, irrespective of the varied length of the video clips. Such a representation would allow direct computation of the similarity or distance between two face videos without the need for frame-to-frame matching. A straight-

forward 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 strategy may be *average* and *max* pooling. There were some successes applying such pooling strategy in building video face representations. For example, the Eigen-PEP representations [18] takes the average of a part-based representation across different video frames and then conducts a PCA dimension reduction. Other works such as the Video Fisher Vector Faces (VF^2) [21] 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 argue that a good pooling strategy should adaptively weight and select the frame-level features across all frames. 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 previous methods, we neither fix the weights nor rely on any particular heuristics to handcraft them. Instead, we designed a neural network to adaptively calculate the weights at runtime. We named our network the neural aggregation network (NAN), whose coefficients can be trained through supervised learning.

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

Our neural aggregation network is designed to inherit the main advantages of pooling techniques, including the ability of handling arbitrary input size and producing ordering invariant representation. The key component of this network is inspired by the Neural Turing Machine [7] and the Orderless Set Network [27], both of which applied an attention mechanism to organize the input through a memory. This mechanism can take 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. Different from the Orderless Set Network [27], we designed a much simpler network structure for our task of video face recognition, instead of the bulky Long Short Term Memory network (LSTM) [9] used in their paper.

Apart from video level representation, our 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 and obtain a 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 [29], the IJB-A dataset [15], and the Celebrity-1000 dataset [19] compared to the baseline solutions using average pooling and other works. Therefore, our research contributes in the following aspects:

- We proposed a 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 achieved state-of-the-art recognition accuracy over three challenging video face recognition benchmarks.

Last but not least, we shall point out though, that our proposed NAN can serve as a general framework for learning content-adaptive pooling. Therefore, it may also serve for feature aggregation for other computer vision tasks such as image recognition and video event recognition [13, 6, 14, 1]. We leave it as our future work to evaluate its performance against other hand-crafted pooling schemes on these tasks, such as VLAD [13, 6, 14] and scene aligned pooling [1].

2. Neural aggregation network

The whole network of our method is composed of two modules. The feature embedding part is a CNN model for face feature extraction from a simple image. The aggregation module fuses all feature vectors to a fixed-dimension representation, which can be used in any decision model. For face identification, the framework of our method is illustrated in Fig. 1. For face verification, we use a modified version employing a Siamese structure [4].

Our new representation stands on the shoulders of state-of-the-art deep CNN model, and is more powerful by adaptively aggregating all frames in the input video into a compact 128 dimensional representation. In this section, we will first introduce the aggregation and feature embedding modules in Section 2.1 and 2.2, respectively, then present the training strategies in Section 2.3.

2.1. Aggregation module

The aggregation module is designed to take benefits from all frames in a video, potentially containing more discriminative information than a single image, and handle arbitrary video size in an unified form, producing an order invariant representation.
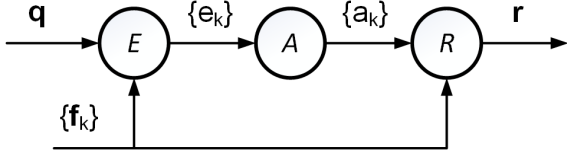
Figure 2. The attention block. It receives a set of feature vectors and filters each of them independently by a kernel $\mathbf{q}$, yielding a set of scalars $\{e_k\}$. There scalars are then passed to a softmax operator, producing a set of weights $\{a_k\}$. Finally, the input feature vectors are fused via Eq. 1.

Consider the video face identification task training on n pairs of labeled data $(X^i, y_i)_{i=1}^n$, where X^i is a video of varying size t_i: $X^i = \{\mathbf{x}_1^i, \mathbf{x}_2^i, ..., \mathbf{x}_{t_i}^i\}$, in which $\mathbf{x}_k^i, k = 1, ..., t_i$ is the kth frame in the video, and y_i is the corresponding subject ID of X^i. Each frame $\mathbf{x}_k^i$ has a corresponding feature representation $\mathbf{f}_k^i$ extracted from the feature embedding module. For better readability, we omit the upper index where appropriate in the following 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

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

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 let the data itself to help generate better weights. For this purpose, we employed a content based attention mechanism [7] in our new network structure for face set representation.

The structure of the aggregation module is shown in Fig. 1. The crux of this module is two attention blocks. Each of them reads all feature vectors from the feature embedding module, filters them with a kernel $\mathbf{q}$ and generates a fixed length representation $\mathbf{r}$ whose dimension is the same as the feature representation $\mathbf{f}_k$ of a single image. The first attention block is used to obtain a content adaptive filter kernel, which we argue that it is better to be varying for different subjects. This way, we invented an optimized content aware video face aggregation structure.

2.1.1 Attention blocks

The attention block in the aggregation module takes a feature set $\{\mathbf{f}_k\}$ as input, 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 normalized weights $\{a_k\}, \sum_k a_k = 1$. Both of these opera-

tions are described by the following equations, respectively:

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

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

The attention block is modulated solely by a filter kernel $\mathbf{q}$. One key advantage of the attention block is that its output is *invariant* to the input order of $\mathbf{f}_k$. It can be seen from Eq. 2, 3 and 1 that, permuting $\mathbf{f}_k$ and $\mathbf{f}_{k'}$ has no effects on the aggregated representation $\mathbf{r}$. Another appealing property is that the number of inputs $\{\mathbf{f}_k\}$ does not affect the size of output $\mathbf{r}$ which is of the same dimension with a single $\mathbf{f}_k$.

We argue that a content adaptive kernel can produce better results. So we employed two attention blocks, with the first to adapt the second's filter kernel, through a transfer layer:

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

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 kernel of the first attention block $\mathbf{q}^0$ and the coefficients of the transfer layer $(\mathbf{W}, \mathbf{b})$ serve as the parameters of the aggregation module, and can be trained by supervised learning via standard gradient descent.

Note that the roles of the first and second attention blocks are different. The first block which is combined with the transfer layer is responsible for generating an input adaptive filter kernel $\mathbf{q}^1$, while the second is to apply this kernel to produce the final aggregated representation $\mathbf{r}^1$. The attention mechanism is suitable to handle orderless inputs of arbitrary size, which is an expected property both in kernel adaptation and result generation.

2.1.2 Discussion

The structure of the aggregation module looks like an unfolded recurrent neural network (RNN) used in the Orderless Set Network [27] due to the coincidence that there are sequentially two attention blocks working in this module. However, there are at least two differences between the proposed network and the one used in their paper.

Firstly, we use a feed forward network, with no weight sharing nor any evolving internal state. Actually, once trained, the first filter kernel $\mathbf{q}^0$ and coefficients of the transfer layer $(\mathbf{W}, \mathbf{b})$ are fixed. Secondly, adding more attention blocks in the aggregation modules sounds like a reasonable extension at first glance. However, that will introduce more non-linearity which is in doubt whether they are beneficial or not, but guarantee to bring more difficulties in training. On trial, we did not observe any improvements using three or four attention blocks, either with or without weight sharing among multiple transfer layers, using complex LSTM unit or not.

2.2. 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 follow the GoogLeNet [25] network structure, and equip it with the Batch Normalization (BN) technique [12]. The output image features from the GoogLeNet are of 128 dimensions which are then fed into the aggregation module. Consequently, our NAN network produces a 128-d feature representation for each input video. In the rest of this paper we will simply refer to the employed GoogLeNet-BN network as CNN.

2.3. Network training

The proposed network can be trained for both face verification and identification tasks described respectively as follows.

Face identification. Face identification attempts to recognize the identity of a probe face video provided a set of gallery face videos/images with known identities. When training, we optimize the coefficients of our network and a fully-connected prediction layer by minimizing 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. In the testing phase, the identity of a probe video will be determined either by feeding it into the trained network (for close-set tests), or choosing from gallery subjects based on the closest L_2 distances of the aggregated features (for open-set tests).

Face verification. In face verification, pairs of face videos are given and the decisions are made to determine whether each video pair is from the same subject or not. To train our network for this task, we build a *siamese* neural aggregation network structure [4] with two NANs sharing their weights, and minimize the average of the contrastive loss [8]: $l_{i,j} = y_{i,j}\|\mathbf{r}_i^1 - \mathbf{r}_j^1\|_2^2 + (1-y_{i,j})\max(0, m - \|\mathbf{r}_i^1 - \mathbf{r}_j^1\|_2^2)$, 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. In the testing phase, the decision is made by comparing the L_2 distance of the aggregated features with a scalar threshold tuned on the training set.

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, few 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 [26, 23], it appears not practical to collect such amount of face videos. We leave an end-to-end training of the NAN as our future work.

3. Experiments

This section evaluates the performance of the proposed Neural Aggregation Network for video face recognition tasks. We will begin with introducing our training details and the baseline methods for the NAN. Then we will report results on three video face recognition datasets: the YouTube Face dataset [29], the IARPA Janus Benchmark A (IJB-A) [15], and the Celebrity-1000 dataset [19].

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[2], and aligned with the LBR method [22]. The input image size is 224x224. After training, the CNN in the feature embedding module is fixed and we focus on analyzing the effectiveness of the neural aggregation module.

The aggregation module is trained on the video face datasets we use 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 ∼1M images in total only takes approximately 3 minutes on CPU of a desktop PC.

3.2. Baseline Methods

The performance of our method is evaluated against a few baseline methods. Since our goal is to build a fixed-size compact representation for video faces, we mainly compared our results with aggregation strategies such as average pooling. Moreover, we also compared it with some set-to-set similarity measurements leveraging pairwise comparison on the image level.

CNN+Mean L_2 measures the similarity of two video faces via averaging the feature distances of all frame pairs. It necessitates storing all image features of a video or sub-
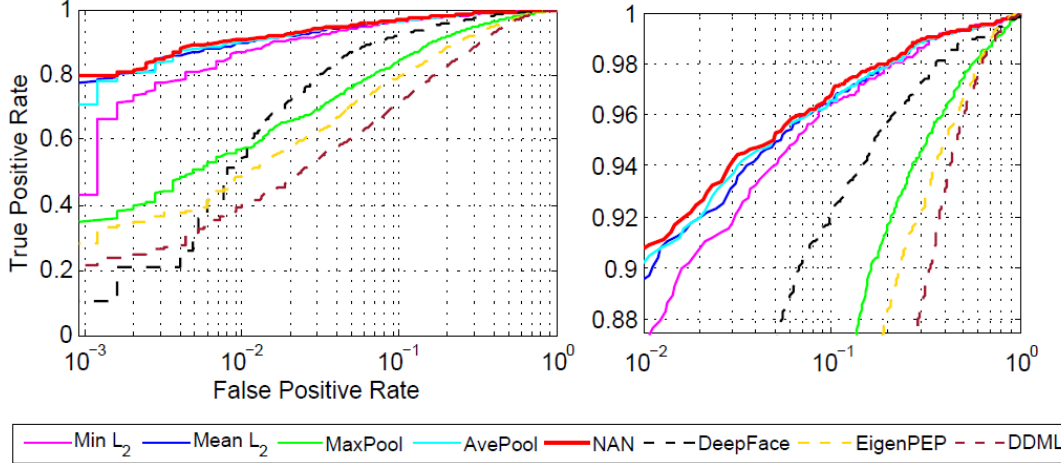
Figure 3. Average ROC curves of different baseline methods and the proposed NAN method on the YouTube Face dataset.

Table 1. Verification accuracy comparison of state-of-the-art methods, our baselines and NAN network on the YouTube Face dataset.

Method	Accuracy (%)	AUC
LM3L [11]	81.3 ± 1.2	89.3
DDML (combined) [10]	82.3 ± 1.5	90.1
EigenPEP [18]	84.8 ± 1.4	92.6
DeepFace-single [26]	91.4 ± 1.1	96.3
DeepID2+ [24]	93.2 ± 0.2	–
FaceNet [23]	95.12 ± 0.39	–
CNN+Min. L_2	94.46 ± 0.10	98.3
CNN+Mean L_2	95.30 ± 0.08	98.6
CNN+MaxPool	86.60 ± 0.13	94.1
CNN+AvePool	95.10 ± 0.08	98.5
NAN	$\mathbf{95.52 \pm 0.06}$	**98.7**

ject, and has $O(n^2)$ complexity for similarity computation.

CNN+Min L_2 is similar to the above but uses the smallest pairwise distance.

CNN+AvePool is average-pooling along each feature dimension for aggregation.

CNN+MaxPool is max-pooling along each feature dimension for aggregation.

CNN+AvePool, CNN+MaxPool and our NAN all produce a 128-d feature representation for each video (or subject of multiple videos), and compute the similarity (L_2 distance) in $O(1)$ time.

3.3. Results on YouTube Face dataset

We first tested our method on the YouTube Face 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. As facial landmarks are not provided in this dataset, we first apply the LBR method [22] to locate the landmarks and align every image before image-level feature extraction with CNN.

The results of our NAN, its baselines and other methods are presented in Table 1, with their ROC curves shown in Fig. 3. It can be observed that our baselines, except CNN+MaxPool, achieve similar accuracies to the state-of-the-art method FaceNet [23], which has an accuracy of 95.12%±0.39. 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 [23]. Our NAN outperforms all the baselines and other methods. It achieves **95.52%±0.06** accuracy, reducing the error of FaceNet by 8.2% and exhibiting a much smaller variance.

Though our NAN has surpassed all the baselines and previous methods on the YouTube Face dataset, we believe that the margin to the baseline performances could be larger. This is because the face variations in the videos of the YouTube Face dataset are relatively small, thus no much beneficial information can be extracted compared to naive average pooling or computing mean L_2 distances. This motivates us to test the network on more challenging datasets such as the IJB-A dataset and the Celebrity-1000 dataset.

3.4. 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

Table 2. Verification accuracy comparison among state-of-the-art methods, our baselines and NAN on the IJB-A face dataset. The true accept rates (TAR) at three false positive rates (FAR) of 0.001, 0.01, 0.1 are reported.

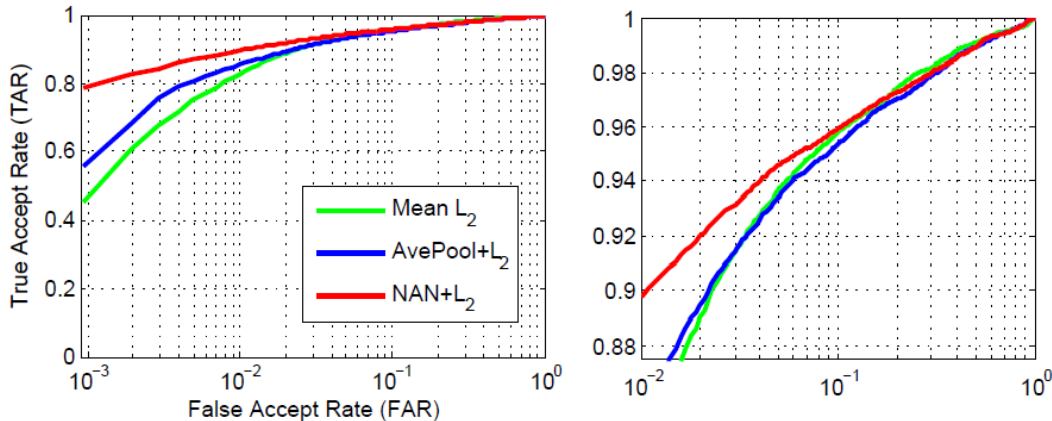| Method	TAR @ FAR=0.001	TAR @ FAR=0.01	TAR @ FAR=0.1
GOTS	0.198 ± 0.008	0.406 ± 0.014	0.627 ± 0.012
OpenBR [16]	0.104 ± 0.014	0.236 ± 0.009	0.433 ± 0.006
LSFS [28]	0.514 ± 0.060	0.733 ± 0.034	0.895 ± 0.013
$DCNN_{manual}$+metric [3]	–	0.787 ± 0.043	0.947 ± 0.011
CNN+Mean L_2	0.453 ± 0.015	0.828 ± 0.023	0.957 ± 0.007
CNN+AvePool	0.559 ± 0.014	0.856 ± 0.017	0.953 ± 0.007
NAN	$\mathbf{0.785 \pm 0.028}$	$\mathbf{0.897 \pm 0.010}$	$\mathbf{0.959 \pm 0.005}$

Figure 4. Average ROC curves of our NAN and the CNN-AvePool, CNN-Mean L_2 baselines on the IJB-A dataset over 10 splits.

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. In this dataset, each training and testing instance, which is called a 'template', comprises a mixture of still images and sampled video frames. The numbers of images in the templates range from 1 to 190 with approximately 10 images per template. In our experiments, the faces are aligned with the provided facial landmarks (two eyes and nose base) before image-level feature extraction with the CNN.

We tested the proposed method on the 'compare' (1:1 matching) protocol for *face verification* on IJB-A. There are 10 training and testing splits. Each of them contains 333 subjects, and its corresponding testing split takes the other 167 subjects. The testing splits contain 11,748 pairs of templates (1,756 genuine and 9,992 imposter pairs) on average. The evaluation metric consists of True Accept Rates (TAR) at different False Accept Rates (FAR).

The results of our method, compared with other methods as well as our baselines are presented in Table 2. THese results show that both NAN and the baselines outperform previous methods such as [28] and [3], and NAN outperforms its baselines especially on the low FAR cases. The TARs of NAN at FARs of 0.001, 0.01 and 0.1 are 0.785,

0.897 and 0.959 respectively. The errors are reduced by about 56%, 51% and 22% respectively compared to the state-of-the-art method [3], or about 51%, 28% and 5% respectively compared to the best results by CNN+Mean L_2 and CNN+AvePool[1]. Note that averaged image features was used by the method presented in [3] for image set representation. Our NAN has learned a discriminative aggregation mechanism for the templates, which is better in verification accuracy than the baseline aggregations and set-distance measurements.

The ROC curves of NAN and its two baselines are presented in Fig. 4. NAN outperforms both baselines with large margins when the FAR is low (<0.03), which suggests that it is more robust and reliable.

3.5. Results on Celebrity-1000 dataset

The Celebrity-1000 dataset is designed to study the unconstrained video-based *face identification* problem. This dataset contains 159,726 video sequences of 1,000 human subjects, with 2.4M frames in total (15 frames per sequence on average). This dataset provides the face regions and 5 facial landmarks from the face detector. We use these landmarks to align the faces before image feature extraction

---

[1]Results by CNN+Min L_2 and CNN+MaxPool are rather poor thus not presented.

Table 3. Identification performance comparison on the Celebrity-1000 dataset with the *close-set* protocol. The rank-1 accuracies (%) are presented.

Method	100 subjects	200 subjects	500 subjects	1000 subjects
MTJSR [19]	50.60	40.80	35.46	30.04
Eigen-PEP [18]	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**

Table 4. Identification performance comparison on the Celebrity-1000 dataset with the *open-set* protocol. The rank-1 accuracies (%) are presented.

Method	100 subjects	200 subjects	400 subjects	800 subjects
MTJSR [19]	46.12	39.84	37.51	33.50
Eigen-PEP [18]	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**

through our CNN module.

Two types of protocols – open-set and close-set – exist on this dataset and we evaluated our NAN on both of them. In the open-set protocol, 200 subjects are used for training, while videos sequences of the rest 800 subjects are used as the gallery set and probe set at testing stage. There are 4 different experimental settings with different number 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.

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 identification scores, and the subject with maximum score is the identification result. We also trained a linear classifier for our baseline method CNN+AvePool to classify each video feature to the subjects. As the features are built on the 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 the subject by aggregating all the 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 L_2 distances between the aggregated probe video representation and each aggregated subject representation.

The identification accuracies of our method on the close-set tests are presented in Table 3. All the baseline methods and our NAN outperformed previous methods by large mar-

gins, and the NAN consistently outperformed the baseline methods on all these tasks. For the 'VideoAggr' approach, NAN reduces the errors of baseline CNN+AvePool by 14%, 9%, 8% and 3% for the settings of 100, 200, 500, and 1000 subjects, respectively. 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 a result consistently better than or on par with the 'VideoAggr' approach, delivering 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 and can generate highly-compact subject level representation.

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 distances between aggregated face representations.

The results in Table 4 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.

4. Closing Remarks

We have presented a Neural Aggregation Network, which is based on CNN and attention mechanism, for video face representation and recognition. It fuses all input frames with a set of content adaptive weights, resulting in a compact (128-d) representation that is invariant to the input frame order, which is an important property in face recognition. The structure of the aggregation module is simple with small computation and memory footprints, but can generate a comprehensive face representation after trained through supervised learning. The experiments have shown than the proposed NAN network consistently outperforms the baselines of hand-crafted aggregations, and it sets new state-of-the-art on the three tested video face recognition 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. In addition, we foresee a further performance boost of NAN when trained in an end-to-end fashion in the presence of enough video data.

References

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

[2] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *Computer Vision–ECCV 2014*, pages 109–122. Springer, 2014.

[3] 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 *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 118–126, 2015.

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

[5] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[6] M. Douze, J. Revaud, C. Schmid, and H. Jégou. Stable hyper-pooling and query expansion for event detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1825–1832, 2013.

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

[8] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 1735–1742. IEEE, 2006.

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

[10] J. Hu, J. Lu, and 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*, pages 1875–1882, 2014.

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

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

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

[14] H. Jégou and A. Zisserman. Triangulation embedding and democratic aggregation for image search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3310–3317, 2014.

[15] 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 *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 1931–1939. IEEE, 2015.

[16] J. C. Klontz, B. F. Klare, S. Klum, A. K. Jain, and M. J. Burge. Open source biometric recognition. In *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, pages 1–8. IEEE, 2013.

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

[18] H. Li, G. Hua, X. Shen, Z. Lin, and J. Brandt. Eigen-pep for video face recognition. In *Computer Vision–ACCV 2014*, pages 17–33. Springer, 2014.

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

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

[21] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman. A compact and discriminative face track descriptor. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

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

[23] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[24] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, pages 2892–2900, 2015.

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

[26] Y. Taigman, M. Yang, M. Ranzato, and 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*, pages 1701–1708, 2014.

[27] O. Vinyals, S. Bengio, and M. Kudlur. Order matters: sequence to sequence for sets. In *Proc. International Conf. on Learning Representation*, San Juan, Puerto Rico, May 2016.

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

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

[30] L. Wolf and N. Levy. The svm-minus similarity score for video face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.