G-FAN: Graph-Based Feature Aggregation Network for Video Face Recognition

He Zhao, Yongjie Shi, Xin Tong, Jingsi Wen, Xianghua Ying, and Hongbin Zha
Key Laboratory of Machine Pereception (MOE)
School of Electronics Engineering and Computer Science
Peking University
{zhaohe97, shiyongjie, xin_tong, wenjingsi}@pku.edu.cn {xhying, zha}@cis.pku.edu.cn

***Abstract*—In this paper, we propose a graph-based feature aggregation network (G-FAN) for video face recognition. Compared with the still image, video face recognition exhibits great challenges due to huge intra-class variability and high inter-class ambiguity. To address this problem, our G-FAN first uses a Convolutional Neural Network to extract deep features for every input face of a subject. Then, we build an affinity graph based on the relationship between facial features and apply Graph Convolutional Network to generate fine-grained quality vectors for each frame. Finally, the features among multiple frames are adaptively aggregated into a discriminative vector to represent a video face. Different from previous works that take a single image as input, our G-FAN could utilize the correlation information between image pairs and aggregate a template of face images simultaneously. The experiments on video face recognition benchmarks, including YTF, IJB-A, and IJB-C show that: (i) G-FAN automatically learns to advocate high-quality frames while repelling low-quality ones. (ii) G-FAN significantly boosts recognition accuracy and outperforms other state-of-the-art aggregation methods.**

I. Introduction

The increasing number of videos captured by both mobile devices and CCTV systems around the world has generated an urgent need for robust and accurate video face recognition algorithm. Video face recognition plays an important role in many practical applications such as visual surveillance, access control, person identification, video search, and so on. The still images are generally taken under controlled conditions and with the subject's cooperation. On the contrary, as shown in Figure 1, video faces are usually moving without directly looking at the cameras, which leads to significant pose variations, low resolution, severe motion blur and increases the recognition difficulty. On the other hand, a video clip consists of multiple frames belong to the same person could provide useful temporal and multi-view information which is profitable for recognition. Therefore, one of the critical challenges in video face recognition is how to effectively combine facial features across multiple video frames, maintaining beneficial while discarding noisy information.

A naive approach would be representing a video face as a set of frame-level face features which extracted by deep neural networks [8], [9], [10], [11]. The information across all frames are comprehensively maintained with such representation. However, in order to compare two video faces, one needs to fuse the matching results across all pairs of frames, which is memory-consuming and inefficient, especially for large-scale recognition tasks. A better solution might be extracting features of each frame and then conducting a certain type of pooling to aggregate the frame-level features into a single video-level representation.

Fig. 1. Example video frames of three subjects sample from YTF [1], IJB-A [2], IJB-C [3] datasets.

Average and max pooling [4], [5] are two of the most commonly used pooling methods, where every face images have equal contribution to the final representation regardless of their quality. However, there are many noisy frames with low facial information content, which may degrade the performance. Considering this problem, NAN [6], FAN [12], and C-FAN [7] are proposed to automatically weight each frame based on their qualities at different levels. Those aggregation strategies improve the video face recognition accuracy and only add negligible computational overhead. However, the relationship between each frame among a video clip has been ignored. A template of face frames should be considered as a whole, those features appeared in most cases should be strengthen and those features appeared only in noisy or low-quality frames should be suppressed. As far as we know, GCN is proved to connect objects and model object-to-object relationships efficiently [13], [14].

To this end, we propose a graph-based feature aggregation network (G-FAN), which consists of an Embedding Network and a Graph Convolutional Network (GCN). We cast video face recognition as a template (a set of images belong to the same person) matching problem. For each frame of a template, the Embedding Network is first used to extract deep features. Then, we build an affinity graph, where each node in the graph represent a face feature extracted from a frame.

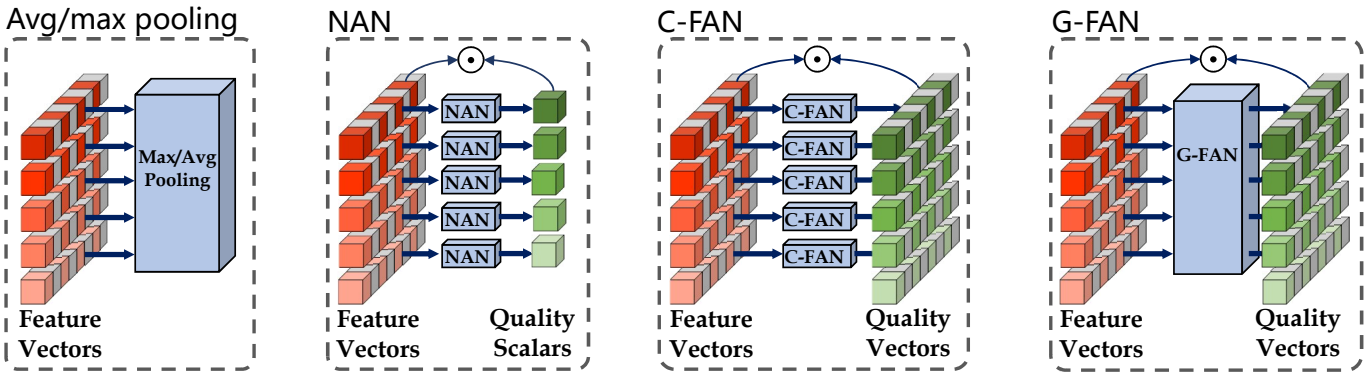

Fig. 2. Comparison of different feature aggregation methods including average and max pooling [4], [5], NAN [6], C-FAN [7] and our G-FAN. The $\odot$ represents element-wise multiplication. Different from NAN and C-FAN which learn quality scalars or vectors separately, G-FAN takes the features of a face template as a whole, and utilize the relationship between each video frames to generate aggregation weights (quality vectors).

Those nodes are connected by there similarity relationship. Taking this graph as input, the GCN would output a set of fine-grained quality vectors and adaptively weights the face features along each dimension among all frames to form a compact and discriminative face representation.

Our network has several favorable properties. First, it is able to tackle the arbitrary number of images for one subject. Second, the aggregation result is invariant to the image order. Third, all parameters of G-FAN are trainable with standard classification or verification loss without any extra supervision signal. The contributions of this paper are summarized as follow:

- We propose the graph-based feature aggregation network, which firstly utilizes correlation information between face frames to aggregate frame features into a single video representation.
- Our G-FAN achieves state-of-the-art performance on three public benchmarks, including YTF [1], IJB-A [2], and IJB-C [3].
- The relationship between visual quality of an image and the quality vector generated by G-FAN is explored by qualitative analysis, which provides an explanation for the effectiveness of G-FAN.

II. Related Work

A. Video Face Recognition

In the past few years, state-of-the-art face recognition methods have been dominated by deep Convolutional Neural Networks (CNNs) [8], [10], [15]. Based on them, several aggregation strategies are adopted for video face recognition. Schroff *et al.* [9] and Taigman *et al.* [11] utilize pairwise frame feature similarity to fuse the matching results but the computation process is time-consuming. Naive average/max-pooling are used in [5], [16] to aggregated features efficiently while ignoring the different frame qualities.

Recently, a few methods take the lead over simple pooling techniques. GhostVLAD [17] employs a modified NetVLAD [18] layer to down weight the contribution of low-quality frames. Yang *et al.* [6] propose NAN, an attention mechanism to adaptively weight the frames and surpass the state-of-the-arts on challenging IJB-A [2] and IJB-B [19] benchmarks. However, these works only consider the instance-level aggregation while our G-FAN learns a fine-grained weight to aggregate the feature vectors in each component separately. As shown in Figure 2, different from other fine-grained aggregation methods C-FAN [7] and FAN [12] where the weight of every image are learned separately, our G-FAN utilize the relationships between the image pairs among the face templates to build an affinity graph and learn quality vectors of each frame by using Graph Convolutional Networks. Therefore, our G-FAN is robust to face templates that have high intra-class variability and outperform other state-of-the-arts on several benchmarks.

B. Graph Convolutional Networks

The generalization of neural networks for arbitrarily structured graphs has drawn great attention in recent years. Different from Convolutional Neural Networks (CNNs), Graph Convolutional Networks (GCNs) are proposed to tackle problems with non-Euclidean data. For the video action recognition task, Wang *et al.* [14] propose to represent videos as space-time region graphs that capture similarity and spatial-temporal relationships. Shen *et al.* [20] utilize GCNs to learn probe-gallery relations for person re-identification. To model dynamic skeletons for human action recognition, Yan *et al.* [21] propose a spatial-temporal GCNs with several types of kernels. Besides, GCNs is also applied for face clustering [22] and consequently improve the face recognition performance. To the best of our knowledge, we are the first to use GCNs to aggregate a template of face images and boost the performance of video face recognition.

III. Graph-Based Feature Aggregation

A. Overview of G-FAN

The overview of the proposed G-FAN is presented in Figure 3. G-FAN incorporates an Embedding Network for feature extraction and a GCN for feature aggregation. The Embedding Network is a CNN pre-trained on MS-Celeb-1M [23] with

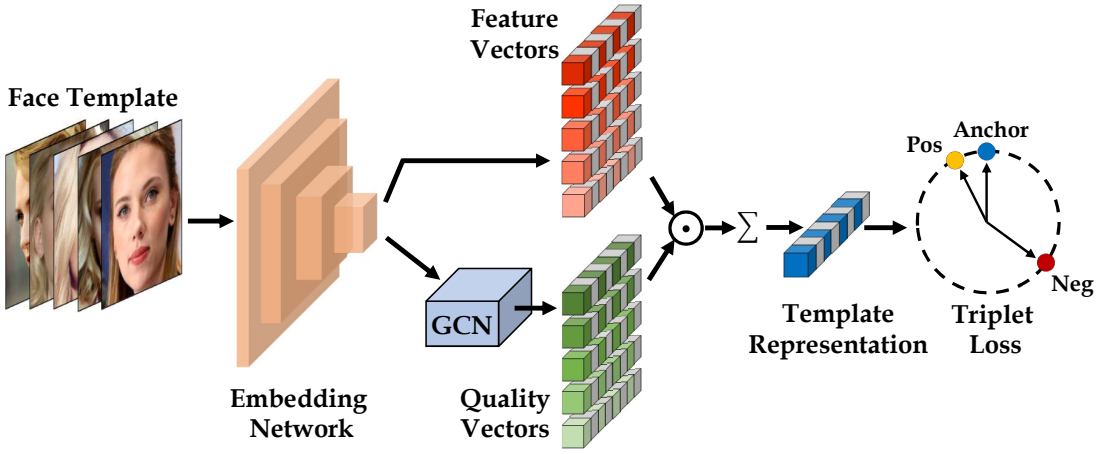

Fig. 3. The overview of our G-FAN, where $\odot$ represents element-wise multiplication and $\sum$ represent sum operation. G-FAN incorporates an Embedding Network for face representation extraction and a GCN for feature aggregation. The aggregated template representation is sent to triplet loss. The whole framework is end-to-end trainable.

Additive Angular Margin loss [8]. Then, we attach the aggregation module to the last layer of the Embedding Network. We jointly train the Embedding Network and feature aggregation module to learn fine-grained quality scores for each feature on the same dataset with triplet loss [9], a different objective function. Finally, the features of a template are pooled with these quality vectors into a single compact vector.

Let $T = \{I_1, I_2, ..., I_N\}$ be a template of face images. Notice that N here is a dynamic number which changes with template size. Let $\mathcal{F}(\cdot)$ denote the feature Embedding Network. The feature vector of i^{th} image in the template is generated by: $\mathcal{F}(I_i) = f_i$, where f_i is a d-dimensional feature vector. In previous works such as NAN [6] and C-FAN [7], the corresponding quality vector q_i of each feature is separately obtained by $\mathcal{Q}(f_i) = q_i$, where $\mathcal{Q}(\cdot)$ represents the feature aggregation module. In NAN [6], the q_i is a unique quality scalar, while for C-FAN [7], q_i is a fine-grained vector which has the same dimension d as the face feature. Different from them, in this paper, we use a GCN as aggregation module $\mathcal{Q}$ which take the whole template of features $F = \{f_1, f_2, ..., f_N\}$ as input and calculate fine-grained quality vectors

$$Q = \mathcal{Q}(F) = \{q_1, q_2, ..., q_N\} \tag{1}$$

for every frames simultaneously. The output Q for the GCN is in the same dimension as the input features F, which is $N \times d$. The details of our GCN will be introduced in the next part.

B. Graph Convolutional Network

Similarity Graph. We measure the similarity between each face representation in the feature space to construct the similarity graph. Specifically, for a template of features $F = \{f_1, f_2, ..., f_N\}$, we build a fully connected graph with N nodes, where each node represents an image feature.

Here, the pairwise similarity between every two features is calculated as $E(f_i, f_j) = cos(f_i, f_j)$, and assigned to each edge. According to previous works [8], [24], after normalization, the deep features are distributed on a hyper-sphere manifold where the cosine distance could be naturally used as a similarity metric. After computing the similarity matrix, we perform normalization on each row of the matrix to help stabilize training. Motivated by recent works [14], [13], we adopt the softmax function as

$$G^{sim}_{i,j} = \frac{\exp\left(s \cdot E(f_i, f_j)\right)}{\sum_{j=1}^{N} \exp\left(s \cdot E(f_i, f_j)\right)}, \tag{2}$$

where s is a hyper-parameter to enhance the difference between each feature. The normalized G^{sim} is taken as the adjacency matrix representing the similarity graph.

Convolutions on Graph. As illustrated in Figure 4, we apply a two-layer Graph Convolutional Network (GCN) [25] to perform reasoning on the graph. Different from standard convolutions that operate on a regular local grid, the graph convolutions allow us to compute the response of a node based on both its feature and its neighbors defined by the graph relations. Formally, we can represent the l^{th} layer of graph convolution as

$$F^{(l+1)} = \sigma\left((G^{sim})^{(l)} F^{(l)} W^{(l)}\right), \tag{3}$$

where G^{sim} represents the similarity graph with $N \times N$ dimensions, $F^{(l)}$ is the input $N \times d$ face features of l^{th} layers, and $W^{(l)}$ is the learnable weight matrix with dimension $d \times d$ in our case. Therefore, the output $F^{(l+1)}$ is still in $N \times d$ dimensions and this operation can be stacked into multiple layers. Besides, we apply σ, a non-linear activation function ReLU before the feature is forwarded to the next layer.

Feature Aggregation. Taking a template of features $\{f_1, f_2, ..., f_N\}$ as input, our GCN outputs quality vectors $\{q_1, q_2, ..., q_N\}$, and the j^{th} component of the i^{th} vector is normalized by:

$$w_{i,j} = \frac{\exp(q_{i,j})}{\sum_{k=1}^{N} \exp(q_{kj})}. \tag{4}$$

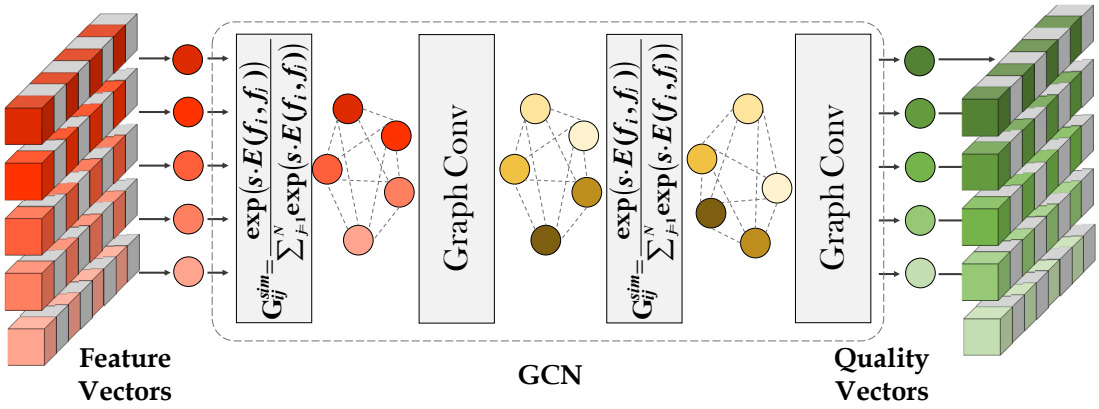

Fig. 4. The details of our GCN. Take template features as input, our GCN utilizes the relation between each frame and calculates the quality vectors to aggregate features into a compact template representation. Note that the adjacency matrix is computed before every convolutional layer.

The aggregated face representation of a template is obtained by pooling the features with the normalized quality vectors:

$$r = \sum_{i=1}^{N} f_i \odot w_i, \tag{5}$$

where $\odot$ represents the element-wise multiplication. The dimension of aggregated feature r is the same as the single face feature extracted by the Embedding Network.

C. Loss Function

To train our proposed G-FAN, we use a template triplet loss, where each triplet consists of three random templates. Those templates are randomly sampled online for every mini-batch and we use Equation 5 to extract template representations. We adopt online hard triplet mining and the loss function can be represented as:

$$\mathcal{L}_{\text{triplet}} = \sum_{i=1}^{M} \max\{0, \alpha + d\left(\mathbf{r}^a, \mathbf{r}^+\right) - d\left(\mathbf{r}^a, \mathbf{r}^-\right)\}, \tag{6}$$

where M is the number of triplets, $\mathbf{r}^a$, $\mathbf{r}^+$ and $\mathbf{r}^-$ are the fused features of anchor, positive and negative templates, respectively. We use a margin parameter α to adjust the convergence difficulty, and $d(\cdot)$ represents the squared euclidean distance.

IV. Experiments

A. Implementation Details

Data Preparation. Our G-FAN is trained on MS-Celeb-1M dataset [23], which contains about 3.8M images of 85k unique identities. We employ a facial landmark detection algorithm MTCNN [26] to detect and align all faces in training and testing. Specifically, each face is cropped from the detected region and aligned to 112 × 112 images with similarity transformation based on the detected five facial landmarks. During the training, the data augmentation is performed by image degradation with the same strategies of GhostVLAD [17].

Training Details. For our Embedding Network, we adopt a modified ResNet-50 structure, which uses an improved residual block [8]. It produces 512-dimension discriminative features for each image. Those features are first normalized to be unit vectors and then fed into the aggregation module. The Embedding Network is pre-trained with Additive Angular Margin Loss [8]. We set batch-size to 256. The learning rate starts from 0.1 and is divided by ten at 150K, 240K iterations. The training process is finished at 300K iterations.

After pre-training, the classification layer is removed, and the GCN is concatenated after the feature extraction layer. The whole network is then trained on the same training dataset using the template triplet loss. We set $\alpha = 0.5, s = 5$. Each mini-batch contains 480 images of 60 templates, eight images per template. The network parameters are optimized using stochastic gradient descent with a learning rate of $1e-3$, a momentum of 0.9, and a weight decay of $5e-3$. The whole network is trained for 25K iterations.

B. Datasets and Protocols

We evaluate G-FAN on three commonly used video face recognition benchmarks: the YouTube Face dataset (YTF) [1], the IAPRA Janus Benchmark A (IJB-A) and more challenging IAPRA Janus Benchmark C (IJB-C).

YTF: The YouTube Face dataset is a video face dataset which contains 3,425 videos of 1,595 different subjects. We follow the *unrestricted with labeled outside data protocol* to report the average accuracy with cross-validation on ten folds of 5,000 video pairs.

IJB-A: IJB-A is a template-based unconstrained face recognition benchmark. It contains wide pose and imaging conditions variations. There are 500 subjects with 5,397 images and 2,042 videos in total and 11.4 images and 4.2 videos per subject on average. We report the true accept rates (TAR) vs. false accept rates (FAR) under *1:1 face verification protocol*.

IJB-C: IJB-C is similar to IJB-A but is more challenging due to its complex pose, resolution changes, and large scale. It consist of 31,334 images and 11,779 videos from 3,531 subjects. There are 19,557 positive matches and 15,638,932

negative matches. Here, we also report TAR at various FAR under *1:1 face verification protocol*.

C. Ablation Study on Template Size

TABLE I
VERIFICATION ACCURACY(%) ON YTF OF OUR G-FAN TRAINED WITH DIFFERENT TEMPLATE SIZES.

Sizes	#Identities	#Images	Accuracy (%)
1	85164	3.80M	97.16
2	83391	3.80M	97.23
4	80430	3.79M	97.69
8	74846	3.75M	**97.98**
16	64322	3.62M	97.72
32	47022	3.21M	97.35
64	21599	1.99M	96.62

Different from traditional methods, G-FAN take face templates as input where each template contains a fixed number of images which belong to the same identity. The larger the template size is, the richer correlation information between face frames will be provided. However, the satisfied identities in the training set which contain enough images will also decrease and lead to worse recognition performance. Therefore, we experiment on YTF to explore the proper template size that strikes a balance. As shown in Table I, the accuracy increases with the template size and reach optimum when each template contains eight images. The oversized template degrades the performance due to the lack of training identities and images.

D. Quantitative Analysis on Public Benchmarks

TABLE II
VERIFICATION ACCURACY(%) ON YTF, COMPARED WITH BASELINE METHODS AND OTHER STATE-OF-THE-ART METHODS.

Method	Backbone	Data	Accuracy (%)
DeepFace [11]	CNN	4M	91.40
DeepID2+ [27]	CNN	300K	93.20
FaceNet [9]	GoogLeNet	200 M	95.52
Center Face [28]	LeNet++	0.7M	94.90
AFRN [29]	ResNet-100	3.2M	97.10
CosFace [24]	ResNet-64	5 M	97.60
ArcFace [8]	ResNet-50	3.8M	**97.30**
ArcFace [8]	ResNet-100	3.8M	**98.02**
NAN [6]	GoogLeNet	3.2 M	95.72
QAN [30]	CNN	5 M	96.17
FAN [12]	ResNet-50	3 M	96.21
C-FAN [7]	Face-ResNet	3.8M	**96.50**
Average	ResNet-50	3.8M	97.26
*C-FAN**	ResNet-50	3.8M	97.53
G-FAN	ResNet-50	3.8M	**97.98**

Besides other state-of-the-art methods, we compare our G-FAN with the other two aggregation methods: *Average pooling* and *C-FAN** on three public benchmarks mentioned above. *C-FAN** represents that we re-implement C-FAN based on our own Embedding Network.

YTF: The results of our G-FAN, the baselines, and other methods on YTF are presented in Table II. It can be seen that the performance of our G-FAN is better than most of the existing face recognition and feature aggregation methods. We achieves a mean accuracy of 97.98% which outperform ArcFace [8] by a large margin while using the same backbone ResNet-50. Furthermore, our result is comparable with ArcFace [8] which utilizes a deeper network ResNet-100 as its backbone. The bridged gap of computational complexity shows the effectiveness of our method. Based on our Embedding Network, the accuracy drops 18.21% when G-FAN is replaced with *C-FAN**, which proves that the graph-based aggregation is more suitable than the instance-level method for video face recognition.

IJB-A: Face images in the IJB-A dataset contains more pose and expression variation. Table III presents the numerical results of different methods. G-FAN outperforms baseline and most of the state-of-the-arts by substantial margins, especially on the low FAR cases. For example, the TAR of our G-FAN at FAR of 0.1% and 1% are 95.97% and 98.64%, which improves the baseline by 3.12% and 1.28%, respectively. The performance of AFRN [29] is better than G-FAN at high FAR. However, it is noteworthy that the backbone of AFRN is a modified ResNet-101, which is much deeper than our ResNet-50. Besides, it performs low-rank bi-linear pooling on the feature maps which is time consuming, while our method utilizes pooled feature vectors which is simple yet effective.

TABLE III
VERIFICATION ACCURACY (%) ON IJB-A BENCHMARK COMPARED WITH BASELINE METHODS AND OTHER STATE-OF-THE-ART METHODS. THE TRUE ACCEPT RATES (TAR) VS FALSE ACCEPT RATES (FAR) ARE REPORTED.

Method	1:1 Verification TAR		
	FAR=0.001	FAR=0.01	FAR=0.1
Crystal Loss [31]	94.80	97.10	98.50
NAN [6]	88.10	94.10	97.80
QAN [30]	89.31	94.20	98.02
M-FAN []	94.44	96.56	98.00
C-FAN [7]	91.59	93.97	-
FAN [12]	93.61	97.28	98.94
GhostVLAD [17]	93.50	97.20	99.00
AFRN [29]	94.90	98.50	**99.80**
Average	92.85	97.36	99.10
*C-FAN**	94.74	97.85	99.27
G-FAN	**95.97**	**98.64**	**99.55**

IJB-C: The IJB-C dataset is an extension of the IJB-A and is more challenging. As shown in the Table IV, the overall trend of the results is similar to IJB-A where our G-FAN achieves significant advantages compared with other methods including AFRN [29]. The results validate the effectiveness of our proposed G-FAN on the large-scale unconstrained face recognition task. Besides, the performance gap between our G-FAN and baseline method on IJB-A and IJB-C benchmarks is larger than YTF. That indicates our G-FAN can extract more beneficial information from those data with significant variations for video face recognition.

E. Qualitative Analysis on Public Benchmarks

Quantitative results have proved the effectiveness of our G-FAN on video face recognition. However, the relationship

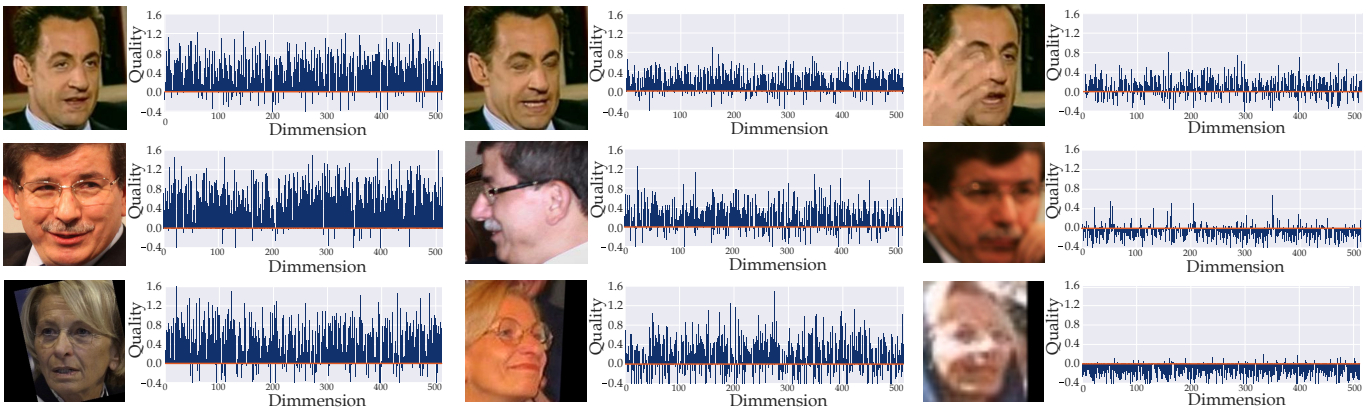

Fig. 5. Quality vectors generated by G-FAN for images of YTF (**top**), IJB-A (**mid**) and IJB-C (**bottom**) datasets. The quality values for each of the 512 components are shown. With different image quality, the distribution of quality values is dynamically changed by our G-FAN.

TABLE IV
VERIFICATION ACCURACY (%) ON IJB-C BENCHMARK COMPARED WITH BASELINE METHODS AND OTHER STATE-OF-THE-ART METHODS. THE TRUE ACCEPT RATES (TAR) VS FALSE ACCEPT RATES (FAR) ARE REPORTED.

Method	1:1 Verification TAR		
	FAR=1e-5	FAR=1e-4	FAR=1e-3
Crystal Loss [31]	87.35	92.29	95.63
CosFace [24]	86.94	91.82	95.37
ArcFace [8]	87.28	92.13	95.55
AdaCos [32]	88.03	92.40	95.65
APA [33]	85.5	92.06	96.23
AFRN [29]	88.30	93.00	96.30
Average	87.16	91.89	95.37
*C-FAN**	87.90	92.57	95.74
G-FAN	**89.42**	**93.83**	**96.38**

between image visual quality and the feature quality vectors generated by G-FAN is yet to be explored. Therefore, we visualize the fine-grained quality vectors of nine images sample from three identities of YTF, IJB-A, and IJB-C datasets in Figure 5.

From the results, we have following observations: (i) The quality scores of high visual quality images are usually higher than images with low visual quality. The images with occlusion, extreme expression and pose, motion blur tend to be assigned with small or even negative quality values. (ii) Different channel of face features encodes the different messages. Therefore, the fine-grained quality vectors of images with low visual quality may also have some high-value dimensions, which help us to utilize all the useful information to aggregate a discriminative template representation. (iii) We observe that faces with noisy labels are usually assigned with low qualities, which show that our G-FAN has the noise-tolerant ability. It is probably due to those noisy images are far from other images on the similarity graph. Further analysis will be discussed in our future work.

V. CONCLUSION

This paper proposes a graph-based feature aggregation network (G-FAN) for video face recognition. G-FAN utilizes correlation information between video frame pairs and adaptively predicts context-aware quality vectors for each deep feature extracted by the CNN face model. We aggregate video face into a compact and discriminative vector and perform efficient recognition with small computation and memory footprints. Experiments on three public video face benchmarks show that our method outperforms other state-of-the-arts. The qualitative analysis demonstrates that G-FAN can maintain the discriminative features while discarding the noisy features. The proposed aggregation scheme can be used for general video or set representation, and we plan to apply it on other vision tasks in our future work.

REFERENCES

[1] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *CVPR*, 2011, pp. 529–534. 1, 2, 4

[2] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain, "Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a," in *CVPR*, 2015, pp. 1931–1939. 1, 2

[3] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney *et al.*, "Iarpa janus benchmark-c: Face dataset and protocol," in *ICB*. IEEE, 2018, pp. 158–165. 1, 2

[4] H. Li, G. Hua, X. Shen, Z. Lin, and J. Brandt, "Eigen-pep for video face recognition," in *ACCV*. Springer, 2014, pp. 17–33. 1, 2

[5] O. M. Parkhi, A. Vedaldi, A. Zisserman *et al.*, "Deep face recognition." in *BMVC*, vol. 1, no. 3, 2015, p. 6. 1, 2

[6] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and G. Hua, "Neural aggregation network for video face recognition," in *CVPR*, 2017, pp. 4362–4371. 1, 2, 3, 5

[7] S. Gong, Y. Shi, and A. K. Jain, "Video face recognition: Component-wise feature aggregation network (c-fan)," *ICB*, 2019. 1, 2, 3, 5

[8] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *CVPR*, 2019, pp. 4690–4699. 1, 2, 3, 4, 5, 6

[9] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015, pp. 815–823. 1, 2, 3, 5

[10] W. Feng, W. Liu, H. Liu, and C. Jian, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. PP, no. 99, pp. 1–1, 2018. 1, 2

[11] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *CVPR*, 2014, pp. 1701–1708. 1, 2, 5

[12] Z. Liu, H. Hu, J. Bai, S. Li, and S. Lian, "Feature aggregation network for video face recognition," in *CVPRW*, 2019, pp. 0–0. 1, 2, 5

[13] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," *ICLR*, 2017. 1, 3

[14] X. Wang and A. Gupta, "Videos as space-time region graphs," in *ECCV*, 2018, pp. 399–417. 1, 2, 3

[15] W. Liu, Y. Wen, Z. Yu *et al.*, "Sphereface: Deep hypersphere embedding for face recognition," in *CVPR*, 2017, pp. 212–220. 2

[16] C. Ding and D. Tao, "Trunk-branch ensemble convolutional neural networks for video-based face recognition," *TPAMI*, vol. 40, no. 4, pp. 1002–1014, 2017. 2

[17] Y. Zhong, R. Arandjelović, and A. Zisserman, "Ghostvlad for set-based face recognition," in *ACCV*. Springer, 2018, pp. 35–50. 2, 4, 5

[18] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," *TPAMI*, pp. 1–1. 2

[19] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen *et al.*, "Iarpa janus benchmark-b face dataset," in *CVPRW*, 2017, pp. 90–98. 2

[20] Y. Shen, H. Li, S. Yi, D. Chen, and X. Wang, "Person re-identification with deep similarity-guided graph neural network," in *ECCV*, 2018, pp. 486–504. 2

[21] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *AAAI*, 2018. 2

[22] L. Yang, X. Zhan, D. Chen, J. Yan, C. C. Loy, and D. Lin, "Learning to cluster faces on an affinity graph," in *CVPR*, 2019, pp. 2298–2306. 2

[23] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *ECCV*. Springer, 2016, pp. 87–102. 2, 4

[24] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *CVPR*, 2018, pp. 5265–5274. 3, 5, 6

[25] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *ICLR*, 2016. 3

[26] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016. 4

[27] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *CVPR*, 2015, pp. 2892–2900. 5

[28] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *ECCV*. Springer, 2016, pp. 499–515. 5

[29] B.-N. Kang, Y. Kim, B. Jun, and D. Kim, "Attentional feature-pair relation networks for accurate face recognition," in *ICCV*, October 2019. 5, 6

[30] Y. Liu, J. Yan, and W. Ouyang, "Quality aware network for set to set recognition," in *CVPR*, 2017, pp. 5790–5799. 5

[31] R. Ranjan, A. Bansal, H. Xu *et al.*, "Crystal loss and quality pooling for unconstrained face verification and recognition," *arXiv preprint arXiv:1804.01159*, 2018. 5, 6

[32] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li, "Adacos: Adaptively scaling cosine logits for effectively learning deep face representations," in *CVPR*, 2019, pp. 10 823–10 832. 6

[33] Z. An, W. Deng, J. Hu, Y. Zhong, and Y. Zhao, "Apa: Adaptive pose alignment for pose-invariant face recognition," *IEEE Access*, vol. 7, pp. 14 653–14 670, 2019. 6