Metadata-based Feature Aggregation Network for Face Recognition

Nishant Sankaran Sergey Tulyakov Srirangaraj Setlur
Venu Govindaraju
Department of Computer Science and Engineering, University at Buffalo
n6, tulyakov, setlur, govind@buffalo.edu

Abstract

This paper presents a novel approach to feature aggregation for template/set based face recognition by incorporating metadata regarding face images to evaluate the representativeness of a feature in the template. We propose using orthogonal data like yaw, pitch, face size, etc. to augment the capacity of deep neural networks to find stronger correlations between the relative quality of the face image in the set with the match performance. The approach presented employs a siamese architecture for training on features and metadata generated using other state-of-the-art CNNs and learns an effective feature fusion strategy for producing optimal face verification performance. We obtain substantial improvements in TAR of over 1.5% at 10^{-4} *FAR as compared to traditional pooling approaches and illustrate the efficacy of the quality assessment made by the network on the two challenging datasets IJB-A and IARPA Janus CS4.*

1. Introduction

Face recognition is described as the problem of classifying faces to particular identities or verifying the possibility that two given faces are of a common identity or not. Over the past few years, face recognition has seen tremendous advances in pushing the state-of-the-art performances to near human [16, 11] and sometimes even surpassing human capabilities [9, 14]. Though these systems have demonstrated exemplary performances leading to the community considering constrained face recognition as generally a solved problem, unconstrained face recognition, however, presents a different challenge.

Unconstrained face recognition attempts to address the fact that many face recognition systems are deployed in settings where there is no control over the conditions under which faces are captured with the possibility of uncooperative subjects. In the unconstrained setting (eg. video surveillance), the goal of face recognition systems is to identify subjects (referred to as probe) from a media collection (referred to as gallery) that may have been compiled previously. The probes and galleries are stored as templates - each of which can constitute one or more face images corresponding to a specific identity. These face images are typically generated through a pipeline of face detection [19], landmark identification [12] and finally alignment. The aligned faces are then transformed into a discriminative representation (such as CNN based features [16, 15]) that is compared with similar representations of other face images to determine if the identities present in the images are the same. Several metrics are employed for the purpose of estimating the similarity of face representations such as the euclidean distance, cosine proximity and even metric learning methods [1, 13].

Matching face templates which are comprised of only single images for the probe and gallery each is relatively straight forward with the use of the above mentioned similarity functions, the most common one being the cosine similarity. However, in the unconstrained datasets like IJB-A [6] and YTF [17], face templates contain multiple images and therefore poses a new challenge of determining how to fuse/pool the face features to a single feature vector representative of the template. Typically the simplest solution is employed - naive average/max pooling [11, 3, 4] of the features to yield the template representation. In recent works, more intelligent solutions using weighted averaging have been proposed [18, 8] where the weights are determined by analyzing the features and evaluating its representativeness.

In this paper we present a new approach for pooling features of a template trained in the context of a face verification task. We use metadata accompanying the face images in the template for the purpose of evaluating the importance of each feature in the aggregation process. Metadata for face images include, but are not limited to, the yaw, pitch, roll of the face in the image as well as other external details such as the size of the face crop, positions of the landmarks, etc. The motivation behind our approach stems from the fact that all previous approaches [18, 8] only consider the features for determining the aggregation weights. Generally speaking, the features are generated by a CNN or other embedding system whose optimization criteria is to map all the face images of an identity to a single distinct cluster with

0-7695-6403-8/18/31.00 ©2018 IEEE
DOI 10.1109/ICB2018.2018.00028

minimal within-class variances (to enhance discriminability) and maximal inter-class variances (to enhance separability). But it becomes evident that, in doing so, this very optimization function restricts the ability of a system to exploit the variances amongst the features to determine optimal relative weights for pooling. Hence we conjecture using additional data/metadata which is unperturbed by the optimization process for generating discriminative features would lead to discovering better aggregation weights.

We use the CNNs described in Ranjan *et al*. [12] and Chen *et al*. [2] to obtain the metadata and features used in our approach. We design a Metadata-based Feature Aggregator Network (M-FAN) which takes as input, features, metadata and an extra parameter called seed weights to produce a weighted feature representation for the template. The seed weights are simply initial weight estimates provided to the network intended as a starting point for the optimization process and the network is trained to transform these seed weights based on the corresponding metadata. This parameter presents the possibility of providing the network with previously hand-crafted weights which it can then fine-tune according to the metadata, thereby boosting the performance as compared to using the hand-crafted aggregation weights. We experiment the model on IJB-A and Janus CS4 datasets and obtain compelling improvements over the previous state-of-the-art approaches and show that the M-FAN model improves the performance of face recognition systems using naive pooling strategies.

This paper is organized as follows. Section 2 reviews other works related to our approach. Section 3 describes the proposed algorithm in detail. Section 4 provides the results of our method on standard datasets. Finally, we conclude the paper in Section 5 with a brief discussion and future research possibilities.

2. Related Work

Remarkable advances have been made in the area of face recognition and verification over the past few years with the advent of deep learning. The VGG-Face model [11] which was one of the earliest deep CNN based approaches for face recognition had significantly improved results on LFW by using a 16 layer convolutional network trained on a large face dataset of 2.6M images of 2622 subjects. Schroff *et al*. [14] presented an approach that learnt a mapping of the face images to a compact euclidean space using the triplet loss formulation which resulted in even better performances. DeepFace [16] introduced by Taigman *et al*. used a DCNN coupled with 3D face alignment where the face pose is normalized by warping facial landmarks to a fixed position and is trained on the resulting face images. Many new approaches also investigate employing metric learning in the form of triplet similarity loss or joint Bayesian metric for arriving at an optimal embedding for face recognition [2, 16]. Masi *et al*. [10] presented Pose-Aware-Models (PAM) that handle pose variability by modeling different poses with separate CNNs. Sankaranarayanan *et al*. [13] proposed using triplet probabilistic embedding to learn a low dimensional embedding using the triplet probability constraints for improving face recognition in the wild. The above mentioned approaches work well under constrained scenarios, however, they usually are not as capable of handling unconstrained face recognition involving templates or sets where large appearance variations are prevalent.

Traditional approaches to feature pooling relied on naive averaging or max pooling and sometimes even going to the extent of having to carefully design "weighting functions" that evaluate the quality of the feature vectors and produce more intelligent weights. Several approaches like [16, 14] merely perform pairwise frame feature similarity comparisons or use naive feature pooling [3, 4]. Neural Aggregation Networks (NAN) [18] by Yang *et al*. introduce an automated approach for generating aggregation weights using a cascaded attention mechanism primed on face features in a template and reported state-of-the-art results on IJB-A and YouTubeFace datasets. Template Adaptation [5] discusses applying a form of transfer learning to the set of media in a template using an SVM loss function. Quality Aware Networks (QAN) proposed by Liu *et al*. [8] assess the quality of each image in a set using a quality generation unit that is coupled beside a feature generation unit to facilitate end-to-end learning of optimal template/set representation. However, all the above methods invariably rely on the feature representation of the face images to assess the quality for aggregation. Hence, in order to overcome these limitations, we propose to use orthogonal features such as metadata, that are not explicitly present in the feature representation and which are learnt using different target objectives, for learning effective feature aggregation methods.

3. Metadata-based Feature Aggregation Network (M-FAN)

We discuss the intuition for proposing the M-FAN model in this section and also elaborate upon our network model and the training approach we implemented for our experiments.

3.1. Overview

The entire objective of the M-FAN model is to function as a feature quality evaluator and produce weights corresponding to the "worthiness" of the feature vector as being a part of the template. Let f_i and m_i be the i^{th} feature vector and corresponding metadata vector in a template. We define an evaluator function h_θ to be a function of the metadata vector, parametrized by θ, producing a weight that qualifies the provided metadata. If T denotes the template vector or

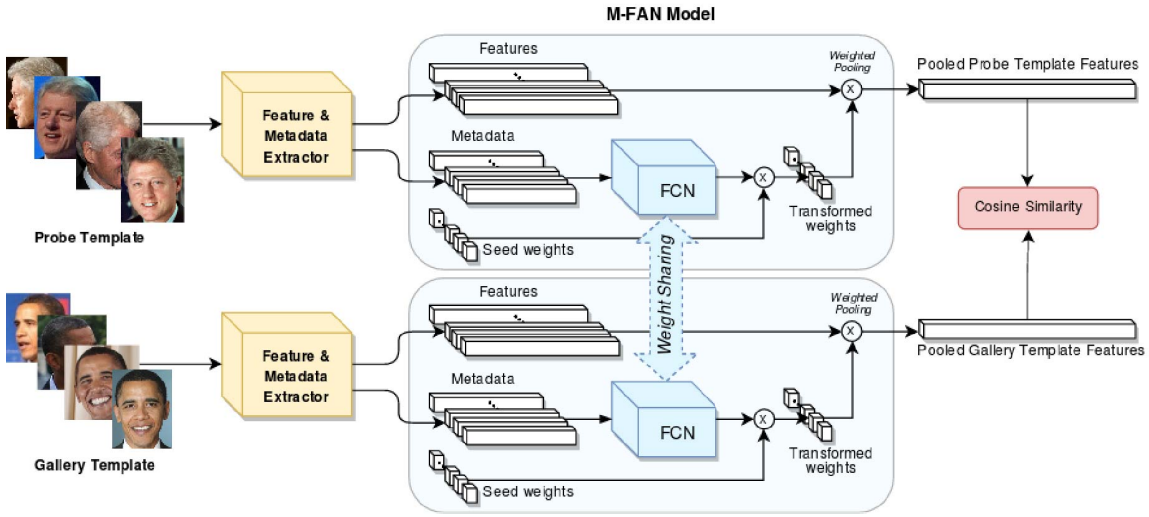


Figure 1: M-FAN architecture. This figure shows the training setup with the M-FAN model deployed as a siamese network. The Feature and Metadata extractor are the networks described in [12] and [2]. The FCN is the only trainable block in the structure.

the pooled features for the template, we have

$$T = \sum_i h_\theta(m_i) f_i \tag{1}$$

Here, h_θ could be realized as any function approximator, and in our case, it is represented by a Fully Connected Network (FCN). The above formulation ensures that the M-FAN network does not rely on the features to make its predictions which is crucial to the performance of our model based on the following reasoning. The feature vectors are typically generated by a face recognizer whose task is to map any and all variations of face images for a particular subject to a single tightly bound cluster in the feature space. It would therefore imply that the feature vectors corresponding to the set of face images for a subject would have minimal variations so as to maximize discriminability for the concerned subject. Now this presents a problem for any aggregation system that attempts to evaluate the relative "richness" of the feature vectors in a template since they would all be extremely similar. This motivates the intuition why the same system would need orthogonal information such as metadata, which is not affected by the feature generation process, to yield context that can help it discriminate between face images of a subject.

Given the template vector construction, the objective of our system then becomes to determine the optimal set of parameters θ that minimizes our cost function defined as:

$$E_{pg} = \left[\frac{T_p . T_g}{\|T_p\| \, \|T_g\|} - Y_{pg} \right]^2 \tag{2}$$

$$Cost = \sum_p \sum_g E_{pg} \tag{3}$$

where T_p and T_g are the probe and gallery template vectors obtained using (1), $Y_{pg} \in [0, 1]$ is the match label for the given probe and gallery templates, E_{pg} is the error in match score prediction and, as is evident, the similarity between the two templates is obtained using the cosine similarity. With these goals in mind, Section 3.2 presents the design of the M-FAN structure.

3.2. Architecture

The setup of the M-FAN architecture is illustrated in Figure 1. The essence of the model is the Fully Connected Network (FCN) that assesses the metadata and outputs a weight for the corresponding feature vector. In practice, the network is also provided a set of seed weights w_i (for example setting $w_i = \frac{1}{n}$, n being the number of images in the template) which it can use as an origin to begin the optimization process. Consequently, the FCN block does not explicitly produce weight predictions as output, rather, produces parameters used to transform the seed weights. Providing seed weights produced by elaborate hand-crafted functions generally improves the ability of the model to converge on better weight predictions.

For training, the M-FAN network is built as a siamese network. The network is provided features, metadata and seed weights for the probe template as the left input and the same for the gallery template as the right input. The features and metadata are obtained from the networks described in [12] and [2]. The FCN block that these inputs go through use shared weights as is typical in siamese ar-

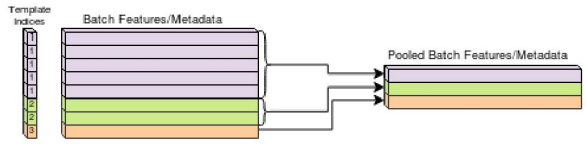


Figure 2: Batch Processing of Templates. The template indices provided indicate which features/metadata comprise a template.

chitectures. The output of the FCN block transforms the seed weights w_i producing w_i' such that $\sum w_i' = 1$, which is then used in conjunction with the corresponding features f_i to create the template vector for each probe and gallery template. The cosine similarity loss layer computes the distance between the templates and is optimized against the match label. During testing, we don't use the siamese setup and instead, present all the inputs for a template to the M-FAN model which produces the aggregated feature vector.

3.3. Gradient backpropagation

The error E_{pg} defined in (2) can be used to derive the gradients for the parameters θ in the FCN being optimized. The gradient for an individual probe gallery template match is computed as:

$$\frac{\partial E_{pg}}{\partial \theta} = \frac{1}{2}\sqrt{E_{pg}}\left[\frac{\|T_p\|\,\|T_g\|\,\frac{\partial T_p.T_g}{\partial \theta} + T_p.T_g\,\frac{\partial \|T_p\|\|T_g\|}{\partial \theta}}{\|T_p\|^2\,\|T_g\|^2}\right] \tag{4}$$

where

$$\frac{\partial T_p.T_g}{\partial \theta} = T_p.T_g' + T_g.T_p' \tag{5}$$

and

$$\frac{\partial\,\|T_p\|\,\|T_g\|}{\partial \theta} = \|T_p\|\,.\,\|T_g\|' + \|T_g\|\,.\,\|T_p\|' \tag{6}$$

follows the product rule of calculus. Similarly, we obtain the gradients of the template vector T and its norm as

$$T' = \sum_i h_\theta'(m_i) f_i \tag{7} \qquad \|T\|' = \frac{T}{\|T\|}.T' \tag{8}$$

The interesting thing to note here is that the gradients for the parameters θ are also a function of the feature vectors f_i. This has a nice effect on the overall training procedure in that even though the FCN block never sees the feature vector for making its predictions, its parameter updates are influenced by f_i, thereby forcing it to learn the implicit correlations between the metadata and features. Moreover, with the presence of only a few dimensions in the input space, as compared to 100s or 1000s when taking the face feature vector also as input, the training algorithm is able to converge faster using fewer network parameters.

3.4. Batch training

During the design of the training setup, it became clear that the network would only be able to train on a single pair of probe and gallery templates at each iteration. This was owing to the fact that each template may be comprised of a variable number of face images, which implies that making batches of probe and gallery templates would be difficult. However, as mentioned in [7], networks generally converge faster and to better minima with batch sizes > 1. In order to work around this problem, we introduced an additional input - indices k_i which held the template indices in the batch that each face image (and the corresponding features and metadata) would be mapped to. This enabled us to group multiple sets of templates as a batch (Figure 2 and to compute the aggregated template vectors using only the corresponding feature vectors as indexed by k_i.

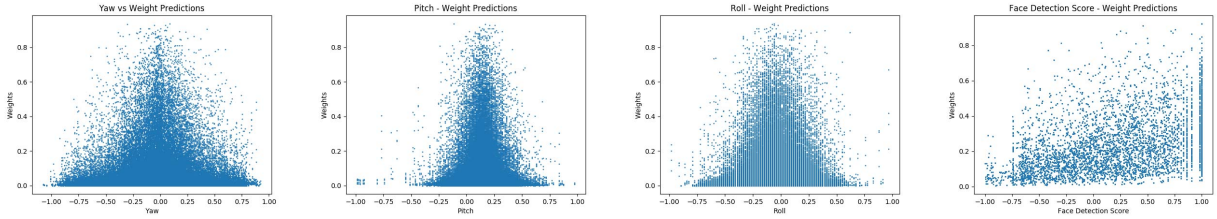
4. Experiments and Analysis

In this section we present the experiment setup and results obtained with our approach on two datasets - IJB-A and JANUS CS4.

4.1. Experiment Setup

The CNN described in [2] produces a 128 dimensional feature vector for each face image. Alongside that, the CNN detailed in [12] produces multiple metadata outputs of which we use yaw, pitch, roll, face bounding box area, gender classification confidence and the face detection score

Figure 3: M-FAN predictions on Janus CS4. The pooling weights are influenced by the orientation of the face, source image size, etc.



(a) Yaw vs Weights (b) Pitch vs Weights (c) Roll vs Weights (d) FD Score vs Weights

Figure 4: Plots showing aggregation weight predictions of M-FAN against various metadata features.

Table 1: IJB-A 1:1 Verification TAR(%)

Method	10^{-1} FAR	10^{-2} FAR	10^{-3} FAR
TE [13]	96.40 ± 0.5	90.00 ± 1.0	81.30 ± 2.0
TA [5]	97.90 ± 0.4	93.90 ± 1.3	83.60 ± 2.7
NAN [18]	97.80 ± 0.3	94.10 ± 0.8	88.10 ± 1.1
M-FAN†	97.97 ± 0.3	96.34 ± 0.3	94.10 ± 0.7
M-FAN‡	$\mathbf{98.00} \pm 0.3$	$\mathbf{96.56} \pm 0.4$	$\mathbf{94.44} \pm 0.5$

TE: Template Embedding, TA: Template Adaptation, NAN: Neural Aggregation Network, † M-FAN (naive), ‡ M-FAN (media)

Table 2: Janus CS4 1:1 Verification TAR(%)

Pooling Method	10^{-2} FAR	10^{-3} FAR	10^{-4} FAR
Naive Average	95.27	90.40	86.54
Media Average	95.38	90.85	86.88
M-FAN (naive)	95.65	90.99	87.35
M-FAN (media)	**95.98**	**92.19**	**88.63**

for our experiments. Our M-FAN model is created with a 4 layer FCN having ReLU as activations. We group the subjects in the datasets into 3 sets - 60% for training (of which 20% is for validation) and the rest for testing. We then used the provided template protocols to generate probe vs gallery matches for the three sets. We train our model on the training set for a maximum of 100 epochs with a learning rate of 0.15 with a decay rate of 0.99 every epoch. We report results on the test set with the model that had the best performance on the validation set. Since the network performance would be influenced by the seed weights provided to it, we conducted 2 sets of experiments - one with the naive average weights and the other by grouping images by their media source and weighting it by the face detection score on each image. For the latter, we first pool all the images corresponding to a particular media source weighted by their face detection scores s_i, i.e., $f_m = \sum_i \frac{exp(s_i)}{\sum_j exp(s_j)} f_i$. The weighted face detection score for the media-pooled images is $s_m = \sum_i \frac{exp(s_i)}{\sum_j exp(s_j)} s_i$. In a similar manner, we aggregate all the media-pooled features f_m with their respective scores s_m to get the template vector T. We record the final weights computed for each image via this method and use them as the seed weights for this experiment which we'll refer to as "media average weights". The models we train on both these experiments are referred to as M-FAN (naive) and M-FAN (media) respectively.

4.2. Results on IJB-A

Here we present the results of the M-FAN model on the IJB-A verification protocol. IJB-A contains 5712 images and 2085 videos of 500 subjects, for an average of 11.4 images and 4.2 videos per subject. We divide the subjects in the provided training split into training and validation (80:20 splits) and evaluate the trained model on the protocol provided in the test splits. The 1:1 verification results are evaluated using the ROC curve and the TAR (True Accept Rate) performance is reported for different FAR (False Accept Rate) values. We present the results reported by the previous state-of-the-art approaches for IJB-A and compare them to our system. The results shown in Table 1 clearly indicate the ability of M-FAN to capture the correlations of the metadata and the features constituting a template and proves its utility as an intelligent aggregation unit.

4.3. Results on Janus CS4

We conduct our experiments on the IARPA Janus Challenge Set 4 (CS4) dataset, which is a superset of the IJB-A dataset [6]; the comparison between CS4 and IJB-A sets is given in [2]. A sample of the weights predicted by M-FAN is shown in Figure 3. Table 2 shows the improvements in performances while using the M-FAN model seeded with naive average weights. It is interesting to note that even when M-FAN is provided naive average seed weights, it is able to perform better than the hand-crafted media average weights. When it is provided the more complex media average weights, it can improve upon its earlier performance by over 1.5%. We also analyzed the weights predicted (during test phase) by the M-FAN model with respect to the various metadata provided to it and plotted the results shown in Figure 4. We can see that the network has learnt to predict low aggregation weights for any orientation that strays far from the frontal pose. Figure 4d depicts the weights for various face detection (FD) scores and here too we see it assign higher confidence to images having higher FD scores.

5. Conclusion

In this paper we discussed about template/set based face verification and how it is deeply influenced by the aggregation or pooling strategy employed in generating representative template features. We presented an approach of using metadata to judge the relative quality of every feature vector in a template for aggregation and investigate its ability to outperform related approaches. Our system produced significant gains over traditional pooling approaches on the IJB-A and Janus CS4 datasets proving the effectiveness of our method. Moreover, our system can be easily plugged into at the end of a face recognition pipeline to optimize the template feature generation process in order to produce improvements in the overall performance.

As part of our future research, we aim to design an end-to-end learning framework for automatically fine-tuning the features and metadata generation networks in view of attaining an optimal template feature generation for face recognition.

6. Acknowledgement

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2014-140716000012. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

References

[1] X. Cai, C. Wang, B. Xiao, X. Chen, and J. Zhou. Deep nonlinear metric learning with independent subspace analysis for face verification. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 749–752. ACM, 2012.

[2] J.-C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep cnn features. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016.

[3] J.-C. Chen, R. Ranjan, A. Kumar, C.-H. Chen, V. M. 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] A. R. Chowdhury, T.-Y. Lin, S. Maji, and E. Learned-Miller. One-to-many face recognition with bilinear cnns. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016.

[5] N. Crosswhite, J. Byrne, C. Stauffer, O. Parkhi, Q. Cao, and A. Zisserman. Template adaptation for face verification and identification. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 1–8. IEEE, 2017.

[6] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1931–1939, 2015.

[7] M. Li, T. Zhang, Y. Chen, and A. J. Smola. Efficient mini-batch training for stochastic optimization. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 661–670. ACM, 2014.

[8] Y. Liu, J. Yan, and W. Ouyang. Quality aware network for set to set recognition. *arXiv preprint arXiv:1704.03373*, 2017.

[9] C. Lu and X. Tang. Surpassing human-level face verification performance on lfw with gaussianface. 2015.

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

[11] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *BMVC*, volume 1, page 6, 2015.

[12] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa. An all-in-one convolutional neural network for face analysis. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 17–24. IEEE, 2017.

[13] S. Sankaranarayanan, A. Alavi, C. D. Castillo, and R. Chellappa. Triplet probabilistic embedding for face verification and clustering. In *Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on*, pages 1–8. IEEE, 2016.

[14] 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.

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

[16] 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.

[17] 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.

[18] J. Yang, P. Ren, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation network for video face recognition. *arXiv preprint arXiv:1603.05474*, 2016.

[19] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.