Attention-Based Template Adaptation for Face Verification

Bin Dong, Zhanfu An, Jian Lin and Weihong Deng
Beijing University of Posts and Telecommunications

Abstract— **In this paper, we propose an Attention-Based Template Adaptation (termed as ABTA) algorithm for face recognition in the unconstrained environment. This ABTA algorithm can be divided into two modules, which consist of an attention-based neural network (feature extractor module) to integrate the template features of various lengths to a single fixed length feature representation according to the attention mechanism, and a template adaptation module (transfer module) which is used to transfer the knowledge of a hold-out dataset to the test templates to improve the performance via transfer learning. The feature extractor module is invariant to the order of the images and videos and can save both memory and computation resources due to its compactness. As for the transfer module, we apply the one-shot similarity to get the scores between the test template pairs, which demonstrates its power in recent research. Our method produces results comparable to the state-of-the-art in the challenging face dataset, IJB-A.**

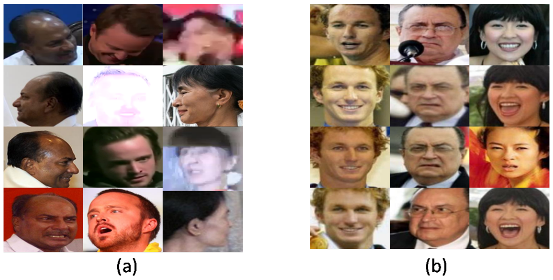

Fig. 1. Face comparison between IJB-A and LFW. The left pictures (a) are from IJB-A, the right pictures (b) are from LFW. Each column represents faces of the same identity. We can see that IJB-A has much more variance than LFW. Faces in LFW have near-frontal bias, while faces in IJB-A have full pose distribution and some of them cannot be detected by Viola Jones Face Detector.

I. INTRODUCTION

Face recognition performance using deep CNN has experienced a significant increase in recent years. Face recognition includes one-to-one comparison, i.e. verification, and one-to-many search, i.e. identification. In face verification, the target is to determine whether the given two faces belong to the same subject, while face identification aims to determine a probe identity based on the one-to-many similarity between a probe media and a gallery of known subjects. In this paper, we mainly focus on face verification task. The verification accuracy rises dramatically mainly due to the novel (usually deeper) network architecture like VGG [14], googlenet [24] or the imaginative new loss, such as triplet loss [17], lifted structured similarity softmax loss [18], center loss [25] etc., as well as multi-task learning such as DeepID2 [20] along with its variants [23], [21] and the recent intriguing All-In-One CNN [15]. Accuracy on certain challenging face datasets with near-frontal bias outperforms or nearly outperforms human performance, such as the YouTube Face [26], and LFW [9]. However, for the more unconstrained setting, the performance degenerates dramatically which means there is still a long way to go for practical application. For example, lots of papers show that their CNNs can achieve over 99% accuracy in LFW [9], but for the much more unconstrained datasets like IJB-A [10], the true accept rates (TAR) when false accept rates (FAR) = 0.01 is only about 80% for its large variance in pose, illumination, expression, aging, cosmetics, races and occlusion. We try to improve the face verification performance in this much more unconstrained setting. The difficulty of IJB-A dataset is manifold.

Firstly, faces in IJB-A are much more unconstrained in head pose, yaw angles, illumination, and expressions, causing large variance within each subject, even larger than the inter-subject difference. To deal with this problem, we modify the template adaptation algorithm proposed in [4] to make the verification process more discriminative. Template adaptation proposed in [4] uses linear SVM in library liblinear [6] and one-shot similarity [27] to calculate the similarity between two templates (or sets of imagery and videos for a subject) to verify them. It improves the performance in IJB-A by a large margin, but it is time-consuming for the large dimension of the VGG features it used and the training of many SVMs.

Secondly, there are several media in each template, so how to compute the similarity of the two templates in face verification task efficiently and effectively is what matters. To cope with this issue, researchers have proposed various ways, including pooling the faces in each template [8], selecting the key face [5] as well as many score-fusion techniques. A straightforward score-fusion method is to calculate the mean similarity score of all pairs between the two templates, which needs $O(n^2)$ complexity. It is burdensome and not optimal for it treats all scores equally. What we pursue is a means to represent a template (or sets of imagery and videos) which promotes the beneficial faces while discarding noisy information at the same time. An interesting method in the recent NAN paper [28] uses an attention-based weighted average algorithm to aggregate the features within each template. However, this method does not consider to project the feature to a more discriminative space, but only does a weighted average of all media.

We learn the essence of these papers and propose an integrated algorithm which performs fairly good on the greatly unconstrained dataset IJB-A while requires less processing time. The rest of the paper is organized as follows. We review some related work in section II. Detail our proposed ABTA

978-1-5090-4023-0/17 $31.00 © 2017 IEEE
DOI 10.1109/FG.2017.116

IEEE computer society

in section III. And show the experimental results in section IV. Finally, in section V we make some conclusion to this paper.

II. Related Work

A. Noble Network Architecture

Noticing that faces in IJB-A have large variance in head poses, yaw angles, illumination, and expressions as shown in Fig.1, some work has been done to solve this problem by using exquisite and delicate new network architecture. [3] uses one intriguing new architecture - the bilinear CNN (B-CNN), which has significantly improved the performance on some fine-grained recognition problems [11]. They apply it to the much more unconstrained dataset, IARPA Janus Benchmark A (IJB-A) [10] and achieved good results. However, the intriguing bilinear CNN is complex for training, which needs plenty of time and resources. Another kind of interesting algorithms using deep multi-pose representations to handle the pose variance is in [1], [12]. In [1], face representation is obtained by passing faces to several pose-specific deep convolutional neural network (CNN) models and get pose-aware features. They generate multiple pose-specific face images by using 3D rendering technique. This ensemble pose-aware CNN is aimed to lessen the large pose change and make face recognition more robust and insensitive to pose change. The idea of [12] is similar to [1], but it uses more sophisticated design of aligning procedure and training details. It uses both 2D in-plane alignment and 3D out-of-plane alignment to align the faces of different poses, then it trains several Pose-Aware CNN Models (PAMs) similar to [1], afterwards it co-trains several PAMs to improve transferability. Also, in [1] they just concatenate the different pose-specific features into one long feature, but [12] fuses the scores obtained by different pose-aware models. We can see that both [1] and [12] need delicate tuning many CNNs and require large data and resources to train, so they are not efficient and time-consuming.

B. Pooling Strategy of Template-Based Verification

As described in section I, another tricky problem in face verification of template (i.e. set of imagery and videos) or video datasets is how to calculate the similarity scores for lots of faces in each template. Given features of faces, a naive method is to calculate the similarity scores for all pairs in the two templates or videos and then perform average (or max) pooling to all the scores to obtain the final score. Obviously, it requires $O(n^2)$ similarity which is tedious when the number of faces is large. In addition, it does not take the information of the faces into consideration, so it is obviously not optimal.

To deal with this issue, different work has been done. [5] proposes a key-face selection method and some feature-level fusion algorithms. They select key-faces based on several criteria, such as entropy, face detection confidence, and the inter-subject variance. They try to select face images which are of good quality and can represent the identity robustly and distinctively. In this way, the number of comparing pairs is reduced markedly, then they do certain kind of feature-level fusion with weighting strategy to get the final feature for each video. Instead of feature pooling, in [8] they choose to use average pooling of face images. They partition a template into several subsets containing faces which share similar appearances and next, they do 3D head pose alignment. Then, a template can be represented by a small collection of 1st order statistics of these subsets of the original template. And a softmax is followed to obtain the final scores. Though interesting, it is also complex, for it includes 3D alignment and image quality estimation and it needs to compute 20 pairs of scores for each template pair. Both [5] and [8] use hand-crafted or heuristic approaches to pool the faces.

Unlike the above algorithm, Yang [28] proposed an adaptive method which can automatically learn the importance of faces in a template and integrate all features of a template to a fixed length representation. It seems that the Neural Aggregation Network (NAN) in [28] strengthens the beneficial faces and suppresses the noisy information. NAN learns the attention-based content-aware weighting factors, which is essential to implement feature-level pooling for faces within a template. Though imaginative, we think that NAN can be further improved by combining transfer learning introduced as follows.

C. Transfer Learning and Template Adaptation

Transfer learning has been well studied in the computer vision community [13]. Transfer learning or domain adaptation is classically applied by pre-training the CNN with certain labeled domain and then replace the final loss layer with new task loss, then fine-tune the CNN in the new domain. The fine-tuning process can either include only the loss layer or both the last few layers and the loss layer depending on the task. Recently, [4] proposed a template adaptation algorithm which only focusses on fine-tuning the last task-specific layer using linear SVMs [6]. They achieve state-of-the-art result in IJB-A by training linear SVMs and utilizing the one-shot similarity.

III. ABTA Algorithm

In this section, we present the attention-based template adaptation (ABTA) algorithm we proposed for face verification. The overall structure of our method is shown in Fig.2. It consists of 5 main steps, that are pose-aware in-plane alignment, feature extraction, triplet probabilistic embedding, attention-based feature aggregation, and template adaptation. And the first 4 steps are the feature extractor module, and step 5 is the transfer module. Our method achieves comparable results to the state-of-the-arts in IJB-A dataset with more efficient representation and less training time.

A. Pose-aware Alignment and Feature Extraction

To handle the large variance in pose, we first do pose-aware in-plane alignment. For the great pose variance in IJB-A as shown in Fig 1, we divide all faces into 4 subsets using their absolute pose angles, that is (0°, 15°), (15°, 45°), (45°,

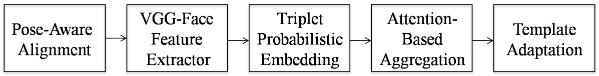
Pose-Aware Alignment
VGG-Face Feature Extractor
Triplet Probabilistic Embedding
Attention-Based Aggregation
Template Adaptation

Fig. 2. Overall procedure. Step 1, align the faces of similar pose using their visible landmarks; step 2, extract the features of the aligned faces using VGG-FACE model; step 3, learn the triplet probabilistic embedding using the training data; step 4, attention-based feature aggregation using neural aggregation network; step 5, template adaptation using one-shot-similarity. Step 1 to step 4 constitute the feature extractor module, and step 5 is the transfer module.

60°), (60°, 90°). We first crop the faces using the ground truth given in the dataset, then we detect the five landmarks (i.e. left eye center, right eye center, nose tip, and two corners of the mouth) using the code given in [22] and we label the landmarks for the faces that can not be detected. Instead of aligning all the faces to the same point and pose, we align them separately based on their respective pose angles. That is to say, we align the faces in each pose subset to fixed points. And if not all five points are visible due to large pose, we only use the visible landmarks to align them to the canonic points. We use 2D in-plane similarity transform rather than 3D alignment for it is difficult to detect as many landmarks as 3D alignment needs and 2D alignment preserves more actual information of the faces without too much loss, but makes verification simpler. Some aligned faces are in Fig.3.

After the alignment, we use the VGG-FACE [14] network to extract the 4096-d features in the penultimate layer. Network training is burdensome and time-consuming. Note that VGG-FACE network is fairly good for face verification for it used over 2.6M faces without alignment, which suits the large variance in IJB-A, we directly extract features using VGG-FACE rather than train a new network.

B. Triplet Probabilistic Embedding

The features extracted from VGG-FACE as described in the previous section are not good enough for two reasons. First, they are not discriminative enough for the training uses only softmax loss which does not emphasize the verification task. As known, features trained using softmax loss focus on the separability between different classes rather than reducing in-class variance. Second, the feature dimension of VGG-FACE is 4096, which is a little large to process. So we determine to do dimension reduction and discriminative embedding together using Triplet Probabilistic Embedding in [16]. With the embedding, it saves both memory and post-processing time significantly.

The idea of TPE is similar to triplet loss [17], but in a probabilistic perspective. Instead of constraining distance, it constrains the similarity directly. Consider a triplet, $t = (v^a, v^p, v^n)$, where v^a and v^p belong to the same class, and v^n belongs to a different class. And they are all aligned features normalized to unit length. We use a, p, n to denote anchor, positive, negative respectively for brevity. And S is a function to calculate the similarity scores between image pairs, such as cosine. We want to do dimension reduction

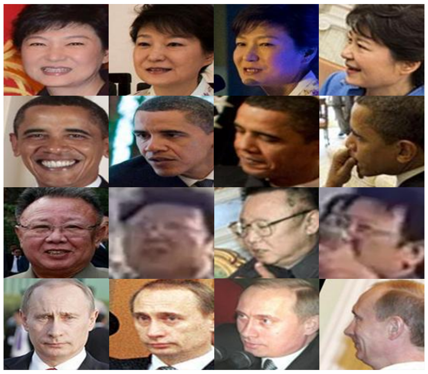

Fig. 3. Pose-aware face alignment. Each row represents faces from a certain identity, and each column represents the aligned faces with the same pose.

and metric learning at the same time. So define

$$S(v^a, v^p) = (Wv^a)^{\mathrm{T}} \cdot (Wv^p) \tag{1}$$

And the following triplet constraint must be satisfied

$$S(v^a, v^p) > S(v^a, v^n) \tag{2}$$

And the probability of each triplet satisfying (2) is

$$P_{apn} = \frac{e^{S(v^a, v^p)}}{e^{S(v^a, v^p)} + e^{S(v^a, v^n)}} \tag{3}$$

Thus, to learn W from some triplets, we solve the following log likelihood optimization problem

$$\arg\min_{W} = \sum_{(v^a, v^p, v^n)} -\log(P_{apn}) \tag{4}$$

We randomly select positive image in the triplet, but select the hardest negative by choosing the negative with the highest similarity to the anchor. The embedding matrix W is initialized with the first 512 primary components of the training data in IJB-A. Then we train W with standard gradient descent. The final embedding feature F_{TPE} can be calculated using

$$F_{TPE}(v_i) = Wv_i \tag{5}$$

We use 512 as the new feature dimension for the first 512 primary components represents 75% variance, which we think as a good initialization of W. We do the TPE using the training data in IJB-A. And it converges quite fast for about 10 minutes for each split in common desktop.

C. Attention-Based Feature Aggregation

As proposed by Yang [23], the Neural Aggregation Network (NAN) utilizes a really simple attention-based approach called attention mechanism [7], [19]. The idea of NAN is quite simple. It aims to compute the weighted average of all faces in a template by its importance to verification. And the importance is learned by data rather than hand-crafted or heuristic methods. It uses two attention blocks to read all feature vectors for each template and then aggregates into a fixed length aggregated feature. Specifically, let f_k be the feature vectors, then an attention block filters them with a

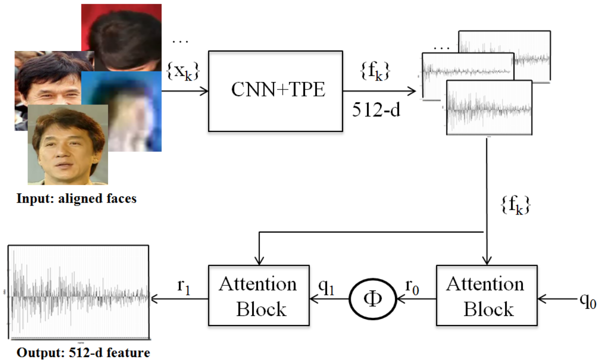

Fig. 4. Attention-based feature aggregation. All pose-aware aligned input faces of one template x_k are fed into VGG-FACE model to extract 4096-d feature, then after TPE, resulting 512-d feature vectors f_k. These representations are then passed to the attention-based feature aggregation part, yielding a compact 512-d feature vector r_1 for each subject. This compact representation can be used for face verification.

kernel q via dot product, yielding a set of corresponding importance e_k. Then e_k is passed to a softmax operator to generate positive weights a_k with $\sum a_k = 1$. Then the weighted average feature r_k can be calculated by the sum of the multiplication of a_k and f_k. These operations can be described by the following equations, respectively.

$$e_k = q^{\mathrm{T}} f_k \tag{6}$$

$$a_k = softmax(e_k) \tag{7}$$

$$r_k = \sum_k a_k f_k \tag{8}$$

We can see that the order of the input of the attention block does not influence the result. To produce a content-adaptive kernel, they apply two attention blocks, with the first to adapt the second filter kernel, through a transfer layer

$$q^1 = \tanh(Wr_0 + b) \tag{9}$$

We can see from above, all the parameters needed in NAN is the first kernel q_0 and the transfer layer parameter W and b. It can be trained using gradient descent. We note that the network is quite simple and the TPE feature used is compact (512-d), so the training process is quite efficient: using comman desktop, training on the training data in IJB-A using CPU only takes about 10 minutes. We apply the attention-based feature aggregation method to make the feature focus on the discriminative information while suppressing the noise. This aggregation is demonstrated in Fig.4.

D. Template Adaptation

Template adaptation [4] is a kind of transfer learning for template-based verification, which combines the CNN features trained in the source domain and the template specific linear SVMs trained on the template using all faces within each templates. Let $f(x)$ be the feature vector of a certain image with 512-d, which is obtained in section III(C). Then we denote one image media as $f(x)$, and one video media as $mean(f(x_i))$, where x_i is the frames in that video. Then a set of media $X = \{f(x_1), f(x_2), ..., f(x_k)\}$ denotes the features within a template. Template adaptation is referred to the problem of max-margin classification of the positive features from a template to the large negative feature set. The positive features are the TPE features F_{TPE} within one of the verify template pair. And the large negative feature set is served as a background feature set, usually the training data in IJB-A. We should note that all the subjects in verify data and training data do not intersect with each other. All features are first normalized to unit length.

More specifically, template adaptation is to train a function $s(P, Q)$ to measure the similarity between the verify template pairs P and Q. First, we train a linear SVM using the unit normalized features of template P as positive features, and the training unit normalized features as large negative features to get SVM_P. In a similar manner, we then train a linear SVM using unit normalized features of template Q as positive features, with training features as negative features, resulting SVM_Q. Then we use the NAN features obtained in the previous section as the average representation of P and Q, denoted as p and q. Then the similarity of template P and Q can be simply calculated by $s(P, Q) = \frac{1}{2}(P(q) + Q(p))$. $P(q)$ means using q to get the classifier margin of SVM_P. So the final score is the average of two classifier margins of $SVMs$ calculated with each other's NAN features. This is the idea of [4], but we combine it with TPE to make it more discriminative and compact, and applying the attention-based feature aggregation rather than simple average feature [4] in the previous section to represent each template more optimaly. We argue that this method is more efficient and effective. The processing time of the tailored template adaptation is about 10 milisecond/template. After applying the tailed template adaptation(TA), our ABTA algorithm is done.

IV. EXPERIMENTS

In this section, we evaluate the ABTA algorithm we proposed in IJB-A dataset. In section IV(A), we briefly introduce IJB-A dataset which is used in our experiments. And in section IV(B), our evaluation pipeline is presented, and in section IV(C), baseline methods and all results are given. The true accept rates (TAR) vs. false positive rates (FAR) are reported. Our method achieves comparable results to the state-of-the-art in IJB-A dataset with more efficient representation and less training time.

A. IJB-A dataset

Our experiments are performed on the IJB-A dataset, a newly published unconstrained datasets. There are 500 subjects in it, and it has both videos and images for each person. In all, the dataset contains 5371 face images and 2042 videos with 25795 unique faces in total. And 5625 templates are formed by the faces. The faces in the IJB-A dataset contain extreme poses and illuminations, much harder than LFW [9]. An additional challenge of the IJB-A verification protocol is that the template comparisons include image to image, image to set and set to set comparisons.

TABLE I

COMPARISON OF OTHER FACE RECOGNITION SYSTEMS ON IJB-A

method/TAR	FAR = 0.001	FAR = 0.01	FAR = 0.1
VGG-Face [14]	0.604(0.06)	0.805(0.03)	0.937(0.01)
PAMs [12]	0.652(0.037)	0.826(0.018)	-
DCNN [2]	-	0.838(0.042)	0.967(0.009)
Pooling-Face [8]	0.631	0.819	-
TPE [16]	0.813(0.02)	0.900(0.01)	0.964(0.005)
All-In-One [15]	0.823(0.02)	0.922(0.01)	0.976(0.004)
TA [4]	0.836(0.027)	0.939(0.013)	0.979(0.004)
NAN [28]	**0.881(0.011)**	**0.941(0.008)**	0.978(0.003)
ABTA-fusion(ours)	0.8191(0.033)	0.9241(0.013)	**0.9807(0.004)**

TABLE II

COMPARISON OF OUR BASELINES ON IJB-A

method/TAR	FAR = 0.001	FAR = 0.01	FAR = 0.1
VGG(ours)	0.6025(0.047)	0.8167(0.025)	0.9545(0.007)
Aligned(ours)	0.6176(0.049)	0.8209(0.024)	0.9569(0.006)
TPE(ours)	0.7068(0.042)	0.8707(0.017)	0.9711(0.005)
NAN(ours)	0.7503(0.039)	0.8837(0.021)	0.9724(0.008)
VGG + TA(ours)	0.8058(0.031)	0.9161(0.012)	0.9758(0.004)
Aligned TA(ours)	0.8064(0.030)	0.9159(0.013)	0.9792(0.004)
TPE + TA(ours)	0.8088(0.025)	0.9177(0.015)	0.9774(0.004)
ABTA(ours)	0.8097(0.027)	0.9182(0.014)	0.9772(0.004)
ABTA-fusion(ours)	**0.8191(0.033)**	**0.9241(0.013)**	**0.9807(0.004)**

In this dataset, each training and testing instance is referred to as a template and comprises a mixture of still images and sampled video frames. Each still image or a set of video frames from the same source is called a media. The numbers of images in the templates range from 1 to 190 with approximately 10 images per template on average. There are 10 training and testing splits. Each of them contains 333 subjects, and its corresponding testing split takes the other 167 subjects. We focus on the face verification task at low false accept rate in this dataset, which is important in practical application. We report the true accept rates (TAR) vs. false accept rates (FAR) and the ROC curves of some results.

B. Evaluation Pipeline

As mentioned in section III, Triplet Probabilistic Embedding (TPE) metric, attention-based network (NAN), and template adaptation (TA) similarity are trained step by step in this paper. First, we divide the faces in each split of IJB-A into four subsets according to their pose angle. Then, we crop the faces using the ground truth given by the dataset. Afterwards, five landmarks (two eyes, nose tip, two mouth corners) are detected using the method in [22]. After that, we align them to four different canonical poses using typical landmarks. If some landmarks are invisible due to large pose, then we only use the visible ones to do the similarity transform. The alignment examples are illustrated in Fig.3. Finally, we resize the pose-aware aligned faces to 224×224, which is the default input size of VGG-FACE, and for simplicity, we just call them aligned features. To compare the effect of alignment, we use the faces cropped by the ground truth without alignment to extract features with VGG-FACE, denoted as VGG feature. After alignment, we apply the TPE method to get 512-d TPE features. Using the TPE features, we trained an attention-based neural network (NAN) as described in section III(D), resulting in the NAN features. Then we apply the learned NAN features as the new mean features in template adaptation and train the template-specific SVMs, and get the final ABTA verification scores. After that, we calculate the TAR at certain points and draw the ROC curves as in Fig.5.

C. Baseline Methods and Results

Our ATBA consists of several phases, so we need to form several baselines to demonstrate the effect of each step. Each previous result is the baseline of the next step. For example, the original VGG feature is the baseline of aligned feature, while aligned feature is the baseline of TPE feature, etc.. Take the TAR@FAR = 0.01 as an example. We denote these baseline using the last steps performed, and they are shown with (ours) in TABLE II. Because we do not know certain implementation details of the methods in other papers, some of the results are different from the reference. For example, in TABLE I, the TAR@FAR=0.01 of VGG-FACE and Template Adaptation in the original papers are 80.5% and 93.9%, while in our implementation as shown in TABLE II, they are 81.67% and 91.61% respectively. As we can see from TABLE II, after alignment, the verification result improves to 82.09% with 0.42% increase. After TPE, it increased sharply to 87.07%. Finally, after applying the delicate NAN and template adaptation which can act as a regularization term, the ABTA result reaches 91.82% which is comparable to the state-of-the-art. We also apply template adaptation to VGG features, aligned features, TPE features, to get the result of VGG + TA(ours), Aligned TA(ours), TPE + TA(ours) respectively as baseline results. We can see from TABLE II, in our implementation, ABTA achieves the best result. We fuse the scores of the four kind of TA (VGG + TA, Aligned TA, TPE + TA, ABTA) in our implementation, and our fusion result is 92.41%, which is the second best result of IJB-A dataset as shown in TABLE I. In Fig.5, we observe that the ROC curve for ABTA-fusion is not very stable when FAR < 1e-3, and we think that it will improve with better fusion method. We can see from TABLE I, our result is only slightly worse than TA [4] and NAN [28] when FAR = 0.001 and FAR = 0.01, but ours achieves state-of-the-arts result when FAR = 0.1.

V. CONCLUSION

We propose the ABTA algorithm which is used to compare face template pairs in unconstrained environment. We achieve comparable result to the state-of-the-art in IJB-A dataset. Our method is really efficient, for we only need about 20 minutes to get the NAN features off-line. Then the

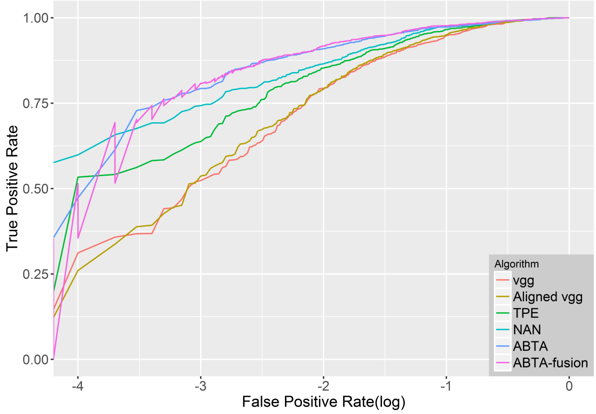

Fig. 5. ROC curve of IJB-A split1

template verification is performed online. It is much faster than the original template adaptation method, for we use much more compact features (512-d vs 4096-d), and we obtained the NAN features beforehand as the mean features used in it. Our method does not need to train large CNN model, which is not necessary and time-consuming. Only a little work can improve the performance significantly. So we believe it is easy to use and implement.

VI. Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (No. 61375031, No. 61573068, and No. 61471048), the Fundamental Research Funds for the Central Universities under Grant No. 2014ZD03-01, This work was also supported by the Program for New Century Excellent Talents in University, Beijing Nova Program under Grant No. Z161100004916088.

References

[1] W. AbdAlmageed, Y. Wu, S. Rawls, S. Harel, T. Hassner, I. Masi, J. Choi, J. Lekust, J. Kim, P. Natarajan, et al. Face recognition using deep multi-pose representations. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

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

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

[4] N. Crosswhite, J. Byrne, O. M. Parkhi, C. Stauffer, Q. Cao, and A. Zisserman. Template adaptation for face verification and identification. *CoRR*, abs/1603.03958, 2016.

[5] N. Damer, T. Samartzidis, and A. Nouak. Personalized face reference from video: Key-face selection and feature-level fusion. In *Face and Facial Expression Recognition from Real World Videos*, pages 85–98. Springer, 2015.

[6] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874, 2008.

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

[8] T. Hassner, I. Masi, J. Kim, J. Choi, S. Harel, P. Natarajan, and G. G. Medioni. Pooling faces: Template based face recognition with pooled face images. *CoRR*, abs/1607.01450, 2016.

[9] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[10] 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 *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1931–1939. IEEE, 2015.

[11] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1457, 2015.

[12] 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.

[13] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[14] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[15] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa. An all-in-one convolutional neural network for face analysis. *arXiv preprint arXiv:1611.00851*, 2016.

[16] S. Sankaranarayanan, A. Alavi, C. D. Castillo, and R. Chellappa. Triplet probabilistic embedding for face verification and clustering. *CoRR*, abs/1604.05417, 2016.

[17] F. Schroff, D. Kalenichenko, and J. W. Philbin. Facenet: A unified embedding for face recognition and clustering. pages 815–823, 2015.

[18] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. pages 4004–4012, 2015.

[19] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. End-to-end memory networks. pages 2440–2448, 2015.

[20] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. pages 1988–1996, 2014.

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

[22] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[23] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. pages 2892–2900, 2015.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. pages 1–9, 2015.

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

[26] 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.

[27] L. Wolf, T. Hassner, and Y. Taigman. The one-shot similarity kernel. In *2009 IEEE 12th International Conference on Computer Vision*, pages 897–902. IEEE, 2009.

[28] J. Yang, P. Ren, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation network for video face recognition. *CoRR*, abs/1603.05474, 2016.