# APA: Adaptive Pose Alignment for Robust Face Recognition

Zhanfu An[1], Weihong Deng[1],Yaoyao Zhong[1], Yaohai Huang[2],Xunqiang Tao[2]
[1]Beijing University of Posts and Telecommunications,
[2]Canon Information Technology (Beijing) Co., Ltd
[1]{anzhanfu, whdeng, zhongyaoyao}@bupt.edu.cn
[2]{huangyaohai,taoxunqiang}@canon-ib.com.cn

## Abstract

*In this paper, we propose a new face alignment method, called adaptive pose alignment (APA) which can greatly reduce the intra-class difference and correct the noise caused by the traditional method in the alignment process, especially in unconstrained settings. Instead of aligning all faces to the pre-defined, uniform frontal shape, we adaptively learn the alignment templates according to the facial poses and then align each face of training or testing sets to its related template. To further improve the face recognition performance, we propose a simple, yet effective feature normalization method which can generate more discriminative feature representation of a face or template combined with the APA method. Furthermore, we introduce a pose-invariant face recognition pipeline that sequentially applies APA based alignment, deep representation by Softmax or Arcface, and the effective feature normalization procedure. We empirically show that APA based images can accelerate the training of deep face recognition model by aligning all the images to the optimal templates. Moreover, experiments show that the proposed method achieves the state-of-the-art performance on challenging IJB-A, IJB-C and CPLFW datasets.*

## 1. Introduction

Face recognition performance using deep CNN has experienced a significant increase in recent years [24] [25] [23][5]. A typical conventional face recognition pipeline usually consists of four stages: face detection, facial landmarks detection and face alignment, feature extraction, and final feature comparison, where face alignment is a crucial step for recognition performance, especially in unconstrained condition with large facial pose. The obtained faces are often different on shape due to factors such as pose, perspective transformation and so on, which will lead to a serious decline on recognition performance. For example, for the same model based CNNs, compared to the accuracy on LFW [10], the accuracy drops about 12%-20% on CPLFW [31] or IJB-A [12]. The main reason for this result is that faces in IJB-A and CPLFW databases are much more unconstrained in head pose, background, and perspective transformation, causing large difference within each subject, even larger than the inter-subject variance. Fig.1 shows some face images sampled from LFW, IJB-A, IJB-C [15] and CPLFW datasets and their corresponding pose distribution (yaw angle). We can see that faces in LFW have near-frontal bias, while faces in other three datasets have full pose distribution and some of them can not be detected by Viola Jones Face Detector[26].

Under this condition, face alignment is an effective approach to alleviate this issue, and further facilitating the recognition tasks [24] [8] [25]. Some works [17] [23] have already indicated that face alignment can efficiently improve the recognition performance. Before the feature extraction step, in both training and evaluation, a better face alignment will decrease the intra-class difference of each subject, further making the classifier more discriminative.

At present, common adopted way to align faces is to use a 2D transformation to calibrate facial landmarks to predefined frontal templates or mean face mode. However, such kind of alignment methods are not optimized under the condition of large pose, which will cause geometric deformation and bring some noise into the image. Furthermore, some researchers have also proposed end-to-end framework that adds alignment to the network, achieved good performance on face recognition. Unfortunately, adding additional structures (e.g. STN [11]) to the network increases the burden on the network and requires longer training time to convergence.

To cope with the above issue, instead of aligning all faces to the pre-defined frontal shape or adding additional architecture to network, we propose a new face alignment method in this paper, namely adaptive pose alignment (APA), to reduce the intra-class difference in order to boost the recognition performance. We adaptively learn the optimal alignment templates according to the facial pose

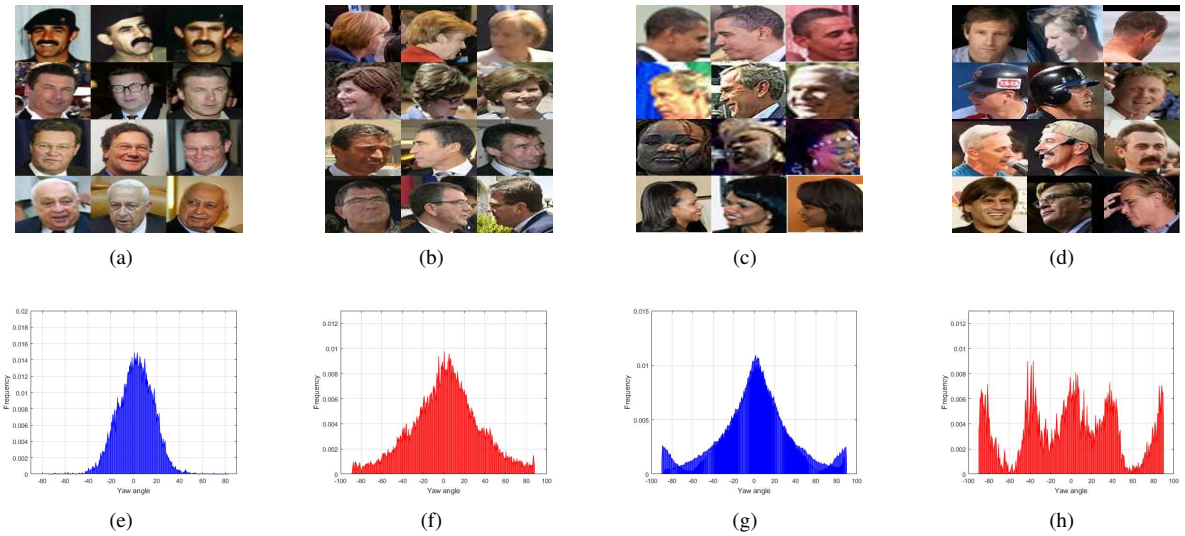



Figure 1. Face images analysis. Top: Sample images sampled from LFW (a), IJB-A (b), IJB-C (c) and CPLFW (d) datasets. Each row presents faces of the same identity. Bottom: Pose distributions (yaw angle) on the four datasets.

and then align each face of training and testing set to its related template, so that the intra-class difference and the noise introduced by alignment process are minimized. Furthermore, we propose a simple, yet effective feature normalization method that combines with the APA, which can further improve face recognition performance. We evaluate the proposed methods on four popular challenging face datasets. And we observe consistent margins, including the IJB-A, IJB-C, and CPLFW datasets, compared to baseline and other competing methods.

In summary, this work contributes to the following aspects:

- We propose a new face alignment method, called APA which can greatly reduce the intra-class difference and correct the noise caused by the traditional method in the alignment process, especially in unconstrained settings. The resulting compact, and yet discriminative face representation enhance the recognition performance of advanced deep face recognition models, even for the recently proposed VGGFace2 model [2] that are known to be invariant to face alignment.
- We empirically show that APA based images can accelerate the training of the deep face recognition model by aligning all the images to the optimal template.
- We propose a pose-invariant face recognition pipeline that sequentially applies APA based alignment, deep representation by Softmax or Arcface [4], and an effective feature normalization procedure. Experimental results show that this pipeline achieves the state-of-the-art performance on challenging IJB-A, IJB-C and CPLFW databases, exceeding the previous state-of-the-art by a large margin.

## 2. Proposed methods

### 2.1. Adaptive Pose Alignment (APA)

Instead of aligning all faces to near-frontal shape, we propose to adaptively learn multiple pose-specific templates as opposed to a single template to align faces: adaptive pose alignment (APA), which indicates reducing the intra-class difference, and preserving the face appearance with little artifact and information loss. The APA consists of three steps: 1) Facial pose estimation; 2) Generating the optimal reference templates based on the pose distribution of the testing dataset. 3) Faces in training or testing dataset are adaptively aligned. Its core is to adaptively generate reference templates that are suitable for face recognition. Then, each face of training dataset or testing dataset are adaptively aligned to its most related reference template.

#### 2.1.1 Template Generation Adaptively

The main purpose of face alignment is to remove the undesired intra-class variability by aligning images to some canonical shapes or configurations. In unconstrained settings, not only do we need to consider intra-class variability, but we also need to reduce the noise caused by the alignment such as the artifacts from similarity transformation or affine transformation. Suppose we define the intra-class difference loss as $d(k)$, the artifact loss of aligned faces as $a(k)$, so the total faces loss of alignment is $Loss(k) = d(k) + a(k)$, where $k$ represents the number of aligned poses. For a training dataset, as the number of aligned templates increases, the intra-class difference $d(k)$ is increasing (Fig.2 left), especially for faces with large pose. But, artifact of aligned faces will become smaller (Fig.2 right). The intra-class

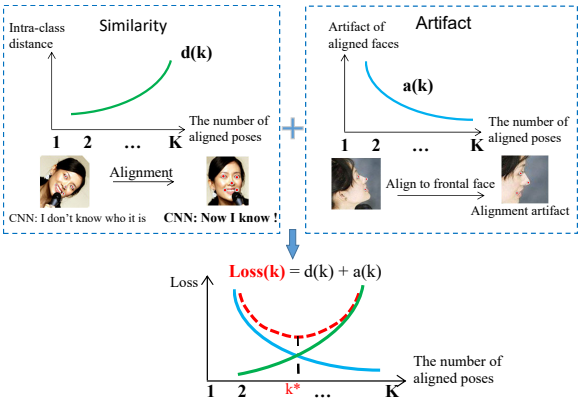



Figure 2. Analysis for large pose face alignment. Top left: As the alignment template $k$ increases, the intra-class distance becomes larger (the intra-class similarity is decreasing). Top right: The alignment template $k$ becomes smaller, the artifacts of aligned face becomes larger, especially when $k = 1$. Bottom: Finding the optimal number ($k^*$) of alignment templates to minimize the total loss of alignment.

similarity is the greatest when all images are aligned to the frontal shape. However, the noise introduced by alignment process is also the largest at this condition. Therefore, we need to find an appropriate number of alignment templates to balance the intra-class difference and information loss, so that the aligned images are most favorable for face recognition (Fig.2 bottom)

$$\begin{aligned} k^* &= arg\min_k Loss(k) \\ &= arg\min_k (d(k) + a(k)). \end{aligned} \tag{1}$$

Differently from approaches that use just a single, frontal template to align all faces [24] [8] [25], our idea is to learn the alignment templates according to pose distribution. Firstly, we need to find a dataset that covers all the possible poses. In each pose, there are as many people as possible that come from different races. Secondly, it is necessary to consider that CNN generalization power is usually proportional to the training data size [13], thus we need to trade-off data partitioning and clustering when determining the number of reference templates $k$. We assume that we have obtained the poses of all faces, and the pose is distributed from $-90°$ to $+90°$. We exploit face symmetry property to mirror all face to one direction of yaw distribution. In this way we can consider only one side of the distribution, for example left side, reducing the number of templates we need to learn. We can cluster all poses $\{\theta\}_i^n$ of reference dataset adaptively using k-means algorithm to find the main $k^*$ templates:

$$\Theta = kmeans(\{\theta\}_i^n)_{k=1}^K, \tag{2}$$

where $\Theta$ represents the collection of clustering centers, the maximum $K$ is 9. The object of K-means clustering is to encourage the sum of distances between features and cluster centers to be the lowest, so that the average loss of transforming faces with different poses to their nearest clustering center template is the lowest. In this paper, we use the elbow method [6] to find the optimak $k$.

### 2.1.2 Pose Estimation

Pose estimation plays an important role in the APA method. In this paper, we calculate the facial pose using the correspondence between the 2D face and the 3D face model.

We use the MTCNN [30] to obtain the 2D labeled landmarks $\mathbf{p}_i = [x_i, y_i]$ of a face image. Then, we mark the corresponding landmarks $\mathbf{P}_i = [X_i, Y_i, Z_i]$ to an aligned 3D generic model. Therefore, a sparse correspondence between 3D and 2D space can be constructed. Weak perspective projection [1] is used to estimate external camera parameters, assuming the principal point in the image center. Finally, we refine the focal length by minimizing landmark reprojection error

$$[\mathbf{p} \quad \mathbf{1}]^T = f\mathbf{A}[\mathbf{R}|\mathbf{t}_{3d}][\mathbf{P} \quad \mathbf{1}]^T, \tag{3}$$

where $f$ is the scale factor, $\mathbf{A}$ is the orthographic projection matrix, $\mathbf{R}$ is the $3 \times 3$ rotation matrix constructed with pitch angle, yaw angle and roll angle, $\mathbf{t_{3d}}$ is the translation vector. From the projection vector $\mathbf{M} = [f, \mathbf{R}, \mathbf{t}_{3d}]$, we extract the rotation matrix $\mathbf{R}$. By decomposing $\mathbf{R}$, we obtain the yaw angle $\theta$ of the face across all the dataset.

### 2.1.3 Alignment Process

Once the optimal alignment templates are determined, all images in dataset are aligned to the template that is associated with it. Given an image of the training or testing dataset, the same method is used to detect the facial landmarks ($\text{lmd}(X)$) and its pose, where $X$ represents a face. And we use the resulting pose to find the optimal alignment template with it

$$R^* = arg\min_i (X - R_i)^2, i = 1, 2, ..., N, \tag{4}$$

where $R$ is reference template, $N$ is the maximum number of reference templates. After finding the optimal template, we mark landmarks on it ($\text{lmd}(R)$). Finally, we seek similarity transformation $T$ to align a face image to the optimal reference template, such that:

$$T^* = arg\min_T \|T[\text{lmd}(X)|\mathbf{1}]' - [\text{lmd}(R^*)|\mathbf{1}]\|_2^2, \tag{5}$$

where $[\text{lmd}(X)|\mathbf{1}]$ is simply an expansion of $\text{lmd}(X)$ by adding an all-one vector $\mathbf{1}$, and $T$ is a homogeneous matrix defined by rotation angle $\theta$, scaling factor $s$, and translation vector $[t_x, t_y]$, as shown in Equ. 6

Figure 3. Face alignment examples using the APA method with $k = 4$. The images from left to right are from LFW, IJB-A, IJB-C and CPLFW datasets. From top to bottom, the pose of the face gradually increases.

$$T = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (6)$$

Through the above process, all faces will be aligned to its related template, and all faces are considered only one side. Fig.3 lists some aligned examples using APA ($k = 4$) sampled from four different datasets.

## 2.2. Feature Normalization

The proposed feature normalization method is based on following two observations. First of all, for most deep networks, the image will be flipped randomly along vertical axes for data augmentation during training. Second, because the face is symmetrical, all faces are aligned to one side when using APA method (Fig.3). In the ideal state, the extracted facial features should also be symmetrical, which provides us a new idea. Suppose we extract the features of left and right faces of a person, so the feature of frontal face of the person can be obtained. We use $f_l$ to represent the feature of the original aligned image, and $f_r$ represents the image feature after mirroring during testing. Therefore, the feature of a face can be expressed as:

$$f = (f_l + f_r)/2. \quad (7)$$

Note that for set-to-set face recognition (for instance IJB-A dataset), $f_l$ and $f_r$ represent a feature vector after template fusion.

In addition, for real world applications, there always exists a dataset bias between the training dataset and testing dataset. In this paper, we use a simple, yet effective method to alleviate the bias for boosting face recognition accuracy, that is, centralizing the facial features that need to be verified or identified to the origin of the reference space during testing. For each feature of testing dataset, we define a new feature to replace the original feature:

$$\overline{f}_i = f_i - \overline{f}_{ref}, i = 1, ..., N, \quad (8)$$

where $\overline{f}_{ref}$ represents the mean feature of the reference data. By doing this, all features are distributed on same reference space. In this space, each feature has the same mean, and comparison between features is more fair.

## 2.3. Pose-invariant Face Recognition Pipeline

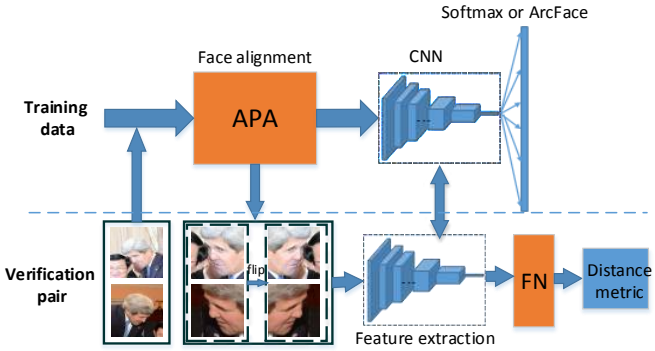



Figure 4. Pose-invariant face recognition pipeline.

The proposed APA method can be used for training dataset or testing dataset. Through experiments, we find that if faces are aligned during training and testing simultaneously, the recognition performance is the best. Therefore, we propose a pose-invariant face recognition pipeline (Fig.4). During training, we first use the proposed APA method to align all the faces in the training dataset. Then, all aligned images are used to train deep network, for example VGG or ResNet network. The loss function can choose Softmax or Arcface. After the model is trained, we can evaluate it using the same aligned images. During the testing phase, we first align the images of probe set in the same manner. We then flip the corresponding images along the vertical axes. Thus, everyone will obtain two aligned images. And we fed them into the trained network to extract features. Finally, the features are normalized using the proposed feature normalization method. The normalized features are use to calculate the similarity of the two faces.

## 3. Experiment and Evaluation

### 3.1. Models and Training Details

The proposed method can be used on any CNNs based face recognition. In this paper, we choose two classic networks to evaluate our method, SE-ResNet-50 [9] (SENet50 for short) and LResNet100-IR [4]. To the best of our knowledge, Cao et al [2] achieved the state-of-the-art results on IJB-A dataset using SENet50 and VGGFace2 dataset [2]. LResNet100-IR is an advanced version of the ResNet network [7], proposed by Deng et al [4], which get state-of-the art performance in the MegaFace Challenge. VGGFace2 [2] is selected as training dataset, which contains over nine thousand identities with between 80 and 800 images for each identity. Loss function uses Softmax and Arcface [4]. Combining the above networks, loss functions, and training data, we mainly train five models, which forms two baselines: **model-A** and **model-C**.

- **model-A** - SENet50 trained with softmax loss, VGGFace2 dataset with no alignment (equivalent to not using the APA method)
- **model-B** - SENet50 trained with softmax loss, APA ($k = 4$) based VGGFace2 dataset
- **model-C** - LResNet100-IR trained with Arcface loss, VGGFace2 dataset with no alignment
- **model-D** - LResNet100-IR trained with Arcface loss, APA ($k = 1$) based VGGFace2 dataset
- **model-E** - LResNet100-IR trained with Arcface loss, APA ($k = 4$) based VGGFace2 dataset

Note that the above models are trained from scratch, and weight initialization uses Xavier method. The mean value of each channel is subtracted for each pixel. Stochastic gradient descent (SGD) is used with mini-batches of size 256 on four GPUs. The initial learning rate is set to 0.1, and this is decreased twice with a factor of 10 when errors plateau.

### 3.2. Data Processing with APA Method

The choice of reference dataset is important for APA method. Inspired by [13], we choose the CASIA-WebFace [29] dataset that contains 10,575 subjects with a total of 494,414 images as our reference dataset. The CASIA-WebFace dataset is a relatively large known public dataset for face recognition. Most of the faces in CASIA-WebFace are centered on the image, which does not require to detect the bounding boxes of faces. Moreover, accurate facial landmarks can be detected. The size of each face is $250 \times 250$.

During both training and testing, we use MTCNN [30] to detect face and landmarks. The bounding box is then extended by a factor of 0.3 to include the whole head and cropped. All cropped faces are resized to $250 \times 250$, which is consistent with the size of the reference template. For faces that MTCNN cannot detect, we use the bounding box and landmarks provided by the dataset. Then, all faces are aligned according to the proposed APA method. Finally, a center region of $200 \times 200$ pixels is cropped from each well-aligned face and resized to fit the input of the network. In this paper, we resize it to $224 \times 224 \times 3$ (SENet50) or $112 \times 112 \times 3$ (LResNet100-IR) respectively (Fig. 3).

### 3.3. Experiments on IJB-A dataset

#### 3.3.1 Janus Benchmark A (IJB-A)

The IJB-A dataset is a publicly available challenging face dataset which contains 500 subjects with a total of 25,791 images (5,396 still images and 20,395 video frames) in total, 11.4 images and 4.2 videos per subject on average. There are 10 training and testing splits. Each training split contains 333 subjects, and its corresponding testing split takes the other 167 subjects.

In all experiments, except for special instructions, results on the IJB-A average over 10 splits. Template encodings are constructed by averaging media encoding over a template, and media encodings are constructed by averaging features across a video [16] [17] [3], then unit normalizing. The similarity between two subjects is computed by the cosine distance.

#### 3.3.2 The Effect of $k$

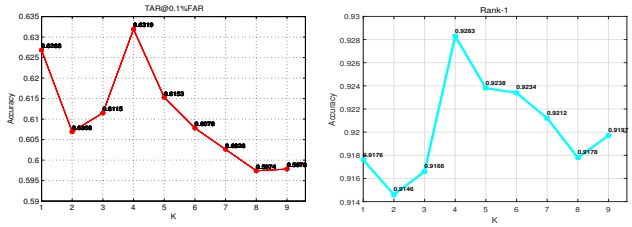
The $k$ is a key in the APA method. In this subsection, we investigate the effect of $k$ through extensive experiments on IJB-A dataset. By varying $k$ from 1 to 9, we use APA method to align nine groups of images of IJB-A dataset. After alignment, we use the off-the-shelf VGG-Face [17] model to extract the 4096-d features of the penultimate layer directly. There are three main reasons for using this trained model. First, training nine models is burdensome and time-consuming. In addition, VGG-Face descriptor has reported fairly good results on the face verification task of LFW benchmark. Moreover, it used over 2.6M faces without alignment to train network, which is suitable for challenging face datasets, like IJB-A. Therefore, we directly extract features using VGG-Face model rather than training new network. Fig. 5 shows the comparison of the recognition results on TAR@FAR=0.1% and Rank-1. We can see that when $k = 4$, both two results achieve the best performance.

#### 3.3.3 Effectiveness of APA

**Accuracy on IJB-A.** We evaluate the effectiveness of APA by comparing three groups of experiments. The results are reported in Table1. In the first set of experiments, we directly extract features of IJB-A dataset using VGGFace model [17]. The second and third groups are tested using our trained models. It is obvious from the experimen-

Table 1. Recognition performance comparison on different models for standard 1:1 face verification and 1:N face identification on the IJB-A.

| Model | Method | 1:1 Verification TAR | | | 1:N Identification TPIR | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | FAR=0.001 | FAR=0.01 | FAR=0.1 | FPIR=0.01 | FPIR=0.1 | Rank-1 | Rank-5 | Rank-10 |
| VGGFace [17] | without APA | 0.5869±0.0566 | 0.8154±0.0249 | 0.9576±0.0074 | 0.4575±0.0568 | 0.7030±0.0285 | 0.9098±0.0102 | 0.9681±0.0068 | 0.9806±0.0047 |
| | APA ($k=1$) | 0.6268±0.0489 | 0.8331±0.0207 | 0.9606±0.0066 | 0.4887±0.0572 | 0.7395±0.0239 | 0.9176±0.0093 | 0.9695±0.0055 | 0.9821±0.0041 |
| | APA ($k=4$) | **0.6435±0.0517** | **0.8467±0.0229** | **0.9673±0.0053** | **0.4907±0.0576** | **0.7567±0.0303** | **0.9306±0.0091** | **0.9785±0.0060** | **0.9863±0.0042** |
| SENet + Softmax | without APA - **model-A** | 0.8842±0.0237 | 0.9514±0.0103 | 0.9828±0.0036 | 0.8164±0.0561 | 0.9218±0.0089 | 0.9780±0.0037 | 0.9903±0.0021 | 0.9928±0.0019 |
| | APA ($k=4$) - **model-B** | **0.8993±0.0294** | **0.9710±0.0070** | **0.9936 ±0.0016** | **0.8332±0.0499** | **0.9466±0.0100** | **0.9831±0.0039** | **0.9945±0.0021** | **0.9964±0.0015** |
| LResNet100-IR + Arcface | No alignment - **model-C** | 0.9564 ±0.0093 | 0.9779±0.0040 | 0.9902±0.0015 | 0.9177±0.0431 | 0.9697±0.0066 | 0.9822±0.0032 | 0.9911±0.0023 | 0.9929±0.0022 |
| | APA ($k=1$) - **model-D** | 0.9616±0.0063 | 0.9803±0.0024 | 0.9907 ±0.0017 | 0.9245±0.0718 | 0.9725±0.0040 | 0.9828±0.0037 | 0.9891±0.0022 | 0.9919±0.0026 |
| | APA ($k=4$) - **model-E** | **0.9661±0.0094** | **0.9873±0.0021** | **0.9948 ±0.0018** | **0.9306±0.0347** | **0.9794±0.0038** | **0.9913±0.0033** | **0.9960±0.0024** | **0.9973±0.0001** |




(a) (b)

Figure 5. Recognition performance varies with the $k$ on IJB-A dataset.

tal results that whether using the existing model (VGGFace model) or the retrained model (**model-A**$\sim$ **E**), when using APA($k=4$), the performance on IJB-A is better than k=1 or no alignment. Specially, it achieves TAR $\sim$ 5.7% improvement on FAR=0.001 using VGGFace model and $\sim$ 1.5% using SENet with softmax.

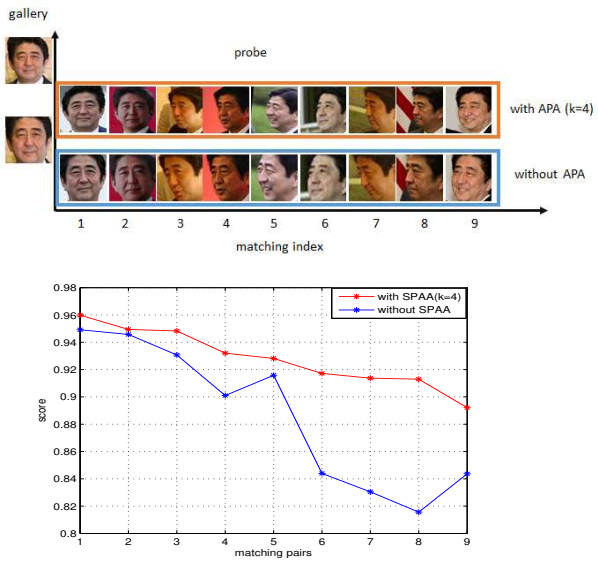
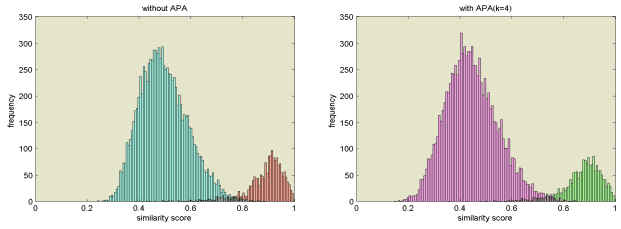



Figure 6. Comparison of intra-class similarity changes. Top: Comparison of image pairs. Bottom: The corresponding value of cosine distance.

**Change of intra-class similarity** The main purpose of the APA is to remove the undesired intra-class variability and increase the intra-class similarity. We design an experiment to verify changes of the intra-class similarity. We select all images of one person whose Subject_ID is 6 from the IJB-A dataset and extract their features using the **model-A** and **model-B**. Then, we calculate the cosine distance between a frontal face and other faces. We select nine groups of images as an example, where their pose is changing from small to large. Fig.6 shows the relationship of similarity changes. It can be seen that after using the APA, the intra-class similarity is significantly improved. Moreover, when the face is more profile, the greater the change is in similarity. The most obvious change of reducing the intra-class variability is that all features of the person will become more compact. It can be seen from the variance between individuals within a group. We first calculate the mean of the features of the same subject $\overline{f} = \frac{1}{N}\sum f_i$. Then, variance is obtained by $var = \frac{1}{N}\sum(f_i - \overline{f})^2$. The variance of the two sets of features is 0.0067 and 0.0052 respectively, which is obviously reduced, illustrating that APA does increase the intra-class similarity.

**Analysis of APA.** Why APA improves the performance of face recognition? The purpose of APA is to reduce the intra-class differences so that the scores of the same sbuject that need to verify pairs become more larger and the scores of the different subjects become more smaller. Thus, the accuracy will be improved. Fig.7 shows the frequency distributions of the similarity scores for face verification on split1 of IJB-A based on **model-A** and **model-B**. From the comparison results, we can see two obvious changes after using APA. First, the scores of the positive pairs are improved. For example, the number of positive pairs with a threshold less than 0.4 is significantly reduced. Second, the scores of the negative pairs are greatly reduced. Thus our APA method does improve pose-invariance for faces with large pose.




(a) (b)

Figure 7. Frequency comparison of similarity scores for set-to-set face verification on split1 of IJB-A dataset. Low scores represent negative sample pairs, and high scores represent positive sample pairs.

Table 2. Recognition performance comparison about feature normalization for standard 1:1 face verification and 1:N face identification on the IJB-A.

| Model | FN | 1:1 Verification TAR | | | 1:N Identification TPIR | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | FAR=0.001 | FAR=0.01 | FAR=0.1 | FPIR=0.01 | FPIR=0.1 | Rank-1 | Rank-5 | Rank-10 |
| VGGFace [17] + APA ($k=4$) | - | 0.6435±0.0517 | 0.8467±0.0229 | 0.9673±0.0053 | 0.4907±0.0576 | 0.7567±0.0303 | 0.9306±0.0091 | 0.9785±0.0060 | 0.9863±0.0042 |
| | √ | **0.7142±0.0318** | **0.8783±0.0201** | **0.9783±0.0056** | **0.6197±0.0461** | **0.8257±0.0172** | **0.9374±0.0094** | **0.9817±0.0052** | **0.9899±0.0037** |
| SENet + Softmax + APA ($k=4$) - **model-B** | - | 0.8993±0.0294 | 0.9710±0.0070 | 0.9936 ±0.0016 | 0.8332±0.0499 | 0.9466±0.0100 | 0.9831±0.0039 | 0.9945±0.0021 | 0.9964±0.0015 |
| | √ | **0.9132±0.0226** | **0.9723±0.0073** | 0.9928 ±0.0014 | **0.8771±0.0294** | **0.9571±0.0066** | **0.9843±0.0041** | **0.9951±0.0014** | **0.9971±0.0017** |
| LResNet100-IR + Arcface + APA ($k=4$) - **model-E** | - | 0.9661±0.0094 | 0.9873±0.0021 | 0.9948 ±0.0018 | 0.9306±0.0347 | 0.9794±0.0038 | 0.9913±0.0033 | 0.9960±0.0024 | 0.9973±0.0001 |
| | √ | **0.9734±0.0061** | **0.9899±0.0018** | **0.9965 ±0.0013** | **0.9442±0.0418** | **0.9834±0.0041** | **0.9928±0.0030** | **0.9968±0.0015** | **0.9977±0.0001** |

Table 3. Performance comparison on the IJB-A with existing methods. '-' implies that the result is not reported for that method. The best results are given in bold.

| Method | 1:1 Verification TAR | | | 1:N Identification TPIR | | | | |
|---|---|---|---|---|---|---|---|---|
| | FAR=0.001 | FAR=0.01 | FAR=0.1 | FPIR=0.01 | FPIR=0.1 | Rank-1 | Rank-5 | Rank-10 |
| VGG-Face [17] | 0.620±0.043 | 0.834±0.021 | 0.954±0.005 | 0.454±0.058 | 0.784±0.024 | 0.925±0.008 | 0.972±0.005 | 0.983±0.003 |
| PAMs [13] | 0.652±0.037 | 0.826±0.018 | - | - | - | 0.840±0.012 | 0.925±0.008 | 0.946±0.007 |
| Masi *et al.* [14] | 0.725 | 0.886 | - | - | - | 0.906 | 0.962 | 0.977 |
| Template Adaptation [3] | 0.836±0.027 | 0.939±0.013 | 0.979±0.004 | 0.774±0.049 | 0.882±0.016 | 0.928±0.010 | 0.977±0.004 | 0.986±0.003 |
| All-In-One+TPE [21] | 0.823±0.020 | 0.922±0.010 | 0.976±0.004 | 0.792±0.020 | 0.887±0.014 | 0.947±0.008 | - | 0.988±0.003 |
| NAN [28] | 0.881±0.011 | 0.941±0.008 | 0.978±0.003 | 0.817±0.041 | 0.917±0.009 | 0.958±0.005 | 0.980±0.005 | 0.986±0.003 |
| $L_2$-softmax [20]+TPE [22] | 0.910±0.013 | 0.951±0.006 | 0.979±0.003 | 0.873±0.024 | 0.931±0.010 | 0.961±0.007 | - | 0.983±0.003 |
| TDFF [27]+TPE [22] | 0.921±0.005 | 0.961±0.007 | 0.989±0.003 | 0.881±0.039 | 0.940±0.009 | 0.964±0.007 | 0.988±0.003 | 0.992±0.003 |
| VGGFace2 (SENet) [2] | 0.904±0.020 | 0.958±0.004 | 0.985±0.002 | 0.847±0.051 | 0.930±0.007 | 0.981±0.003 | 0.994±0.002 | 0.996±0.001 |
| VGGFace2-ft (SENet) [2] | 0.921±0.014 | 0.968±0.006 | 0.990±0.002 | 0.883±0.038 | 0.946±0.004 | 0.982±0.004 | 0.993±0.002 | 0.994±0.001 |
| GridFace [32] | 0.921±0.008 | 0.839±0.014 | - | - | - | 0.929±0.010 | 0.962±0.005 | - |
| Ranjan et al.[18] | 0.952 | 0.969 | 0.984 | 0.92 | 0.962 | 0.975 | 0.986 | 0.989 |
| ArcFace [4] (**model-C**) | 0.9564±0.0093 | 0.9779±0.0040 | 0.9902 ±0.0015 | 0.9177±0.0431 | 0.9697±0.0066 | 0.9922±0.0032 | 0.9911±0.0023 | 0.9929±0.0022 |
| VGG-Face [17] + APA(k=4) | 0.7142±0.0318 | 0.8783±0.0201 | 0.9783±0.0056 | 0.6197±0.0461 | 0.8257±0.0172 | 0.9374±0.0094 | 0.9818±0.0052 | 0.9899±0.0037 |
| **model-B** - APA(k=4) | 0.9132±0.0226 | 0.9723±0.0073 | 0.9928 ±0.0014 | 0.8771±0.0294 | 0.9571±0.0066 | 0.9843 ±0.0041 | 0.9951 ±0.0014 | 0.9971 ±0.0017 |
| **model-E** - APA(k=4) | **0.9734±0.0061** | **0.9899±0.0018** | **0.9965±0.0013** | **0.9442±0.0418** | **0.9834±0.0041** | **0.9928±0.0030** | **0.9968±0.0015** | **0.9977±0.0001** |

#### 3.3.4 The Effect of Feature Normalization

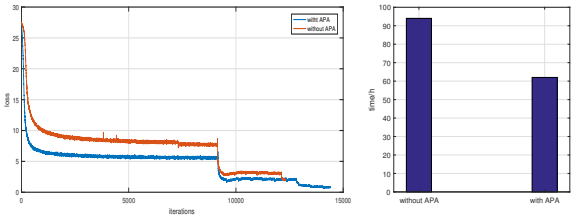
Further improvement is obtained by applying our proposed feature normalization method. In this experiment, the mean feature of IJB-A training set of each split is selected as the central reference feature $\overline{f}_{ref}$. In table 2, we report the improvement for three types of CNN models that we used (VGGFace [17], **model-B** and **model-E**). This table shows that significant improvement in performance is given by the feature normalization.

#### 3.3.5 Training Time

The APA method can not only improve the quality of trained CNN based face recognition model, but it also shorten the training time of the CNN. In this section, we compare the training process of the **model-C** and **model-E**. Fig.8 shows the loss curve and training time comparison of the two models. They are trained with a mini-batch size of 256 on four GPUs. We start with a learning rate of 0.1, divide it by 10 at 9K and 13K iterations, and terminational iteration is set to 14.6K. Note that the convergence of the **model-C** is very slow, so we only iterate 12k. It has been shown that using APA method converges much faster than randomly cropping images method (no alignment), which suggests that a good alignment method can simplify the optimization. Specially, The training time of **model-E** is 1.5 times faster than the **model-C**.

#### 3.3.6 Comparison with State-of-the-art Results

In Table 3, we report a comparison with the state-of-the-art results on IJB-A dataset. It is clear to see that we have significantly improved the recognition rate on IJB-A verification and identification and obtain the state-of-the-art results. Specially, it achieves TAR 97.34% at FAR=0.001, which improve over [2] of about 5.4% and about 3% compared to the state-of-the-art result [4].




(a) (b)

Figure 8. Comparison of loss and training time of LResNet100-IR network trained with VGGFace2 dataset using two different data processing methods: randomly cropping faces and using the APA method to align faces. (a) Loss curve changes. (b) comparison of training time.

### 3.4. Experiments on IJB-C Dataset

The IJB-C dataset [15] is an extension of the IJB-A dataset [12]. The IJB-C dataset contains 3,531 unique subjects with a total of 31,334 still images (21,294 face and 10,040 non-face), averaging to $\sim$ 6 images per subject, and 117,542 video frames collected in unconstrained settings, averaging to $\sim$ 33 frames per subject and $\sim$ 3 videos per subject. Included with the protocols are two disjoint galleries, gallery 1 (G1) and gallery 2 (G2). These galleries are disjoint from each other so that opened-set identification scenarios can be tested. Since the dataset contains two set of galleries G1 and G2, we report the average performance of both the gallery sets.

We first use the trained model, provided by Cao et.al [4] to test on IJB-C dataset. The ResNet-50 (with and without Squeeze-and-Excitation blocks [9]) is trained on

Table 4. 1:1 Face Verification Evaluation on IJB-C.

| Method | IJB-C 1:1 Verification TAR | | | | | | |
|---|---|---|---|---|---|---|---|
| | FAR=$10^{-7}$ | FAR=$10^{-6}$ | FAR=$10^{-5}$ | FAR=$10^{-4}$ | FAR=$10^{-3}$ | FAR=$10^{-2}$ | FAR=$10^{-1}$ |
| GOTS [15] | - | 0.030 | 0.0661 | 0.1467 | 0.3304 | 0.6199 | 0.8093 |
| FaceNet [23] | - | 0.2095 | 0.3330 | 0.4869 | 0.6645 | 0.8176 | 0.9245 |
| VGGFace [17] | - | 0.3220 | 0.4369 | 0.5975 | 0.7479 | 0.8713 | 0.9564 |
| Crystal loss (model-A) [18] | 0.6596 | 0.7646 | 0.8625 | 0.9191 | 0.9572 | 0.9783 | 0.9914 |
| Ranjan et al.[19] | 0.559 | 695 | 0.869 | 0.925 | 0.959 | 0.979 | 0.992 |
| VGGFace2(ResNET-50)[2] | 0.4295 | 0.5191 | 0.6465 | 0.7607 | 0.8649 | 0.9367 | 0.9763 |
| VGGFace2_ft(ResNET-50)[2] | 0.4073 | 0.5644 | 0.6881 | 0.7932 | 0.8866 | 0.9463 | 0.9811 |
| VGGFace2(SENet)[2] | 0.4012 | 0.5353 | 0.6893 | 0.8143 | 0.9019 | 0.9569 | 0.9876 |
| VGGFace2_ft(SENet)[2] | 0.4169 | 0.5320 | 0.6948 | 0.8116 | 0.8995 | 0.9567 | 0.9874 |
| ArcFace [2] (**model-C**) | 0.5996 | 0.6877 | 0.8082 | 0.8873 | 0.9418 | 0.9762 | 0.9923 |
| **model-B** - APA(k=4) | 0.4590 | 0.5741 | 0.7228 | 0.8436 | 0.9257 | 0.9732 | 0.9935 |
| **model-E** - APA(k=4) | **0.6937** | **0.7833** | **0.8550** | **0.9206** | **0.9623** | **0.9842** | **0.9945** |

Table 5. 1:N Face Identification Evaluation on IJB-C.

| Method | IJB-C 1:N Identification TPIR | | | | | |
|---|---|---|---|---|---|---|
| | FPIR=0.001 | FPIR=0.01 | FPIR=0.1 | Rank-1 | Rank-5 | Rank-10 |
| GOTS [15] | 0.0266 | 0.0578 | 0.1560 | 0.3785 | - | 0.6024 |
| FaceNet [23] | 0.2058 | 0.3240 | 0.5098 | 0.6922 | - | 0.8136 |
| VGGFace [17] | 0.2618 | 0.4506 | 0.6275 | 0.7860 | - | 0.8920 |
| Crystal loss (model-A) [18] | **0.7842** | 0.8609 | 0.9191 | 0.9456 | - | 0.9753 |
| Ranjan et al.[19] | - | **0.873** | 0.9255 | 0.949 | 0.9695 | 0.9755 |
| VGGFace2(ResNet-50)[2] | 0.5310 ±0.0166 | 0.6450±0.0228 | 0.7648±0.0237 | 0.8619±0.0175 | 0.9211±0.0108 | 0.9396±0.0088 |
| VGGFace2_ft(ResNet-50)[2] | 0.5852 ±0.0131 | 0.6950±0.0177 | 0.8005±0.0199 | 0.8814±0.0227 | 0.9344±0.0178 | 0.9486 ±0.0138 |
| VGGFace2(SENet)[2] | 0.5671 ±0.0042 | 0.6676±0.0211 | 0.7790 ±0.0259 | 0.8745±0.0181 | 0.9304±0.0101 | 0.9468±0.0101 |
| VGGFace2_ft(SENet)[2] | 0.5728 ±0.0068 | 0.6914±0.0291 | 0.8165 ±0.0207 | 0.8875±0.0208 | 0.9377 ±0.0172 | 0.9515±0.0131 |
| ArcFace [4] (**model-C**) | 0.7019 ±0.0160 | 0.8077±0.0161 | 0.8933 ±0.0197 | 0.9357±0.0161 | 0.9657 ±0.0111 | 0.9748±0.0081 |
| **model-B** - APA(k=4) | 0.5906±0.0205 | 0.7206 ±0.0197 | 0.8565 ±0.0195 | 0.9146±0.0259 | 0.9554±0.0156 | 0.9671±0.0119 |
| **model-E** - APA(k=4) | 0.7604±0.0319 | 0.8577±0.0339 | **0.9272±0.0208** | **0.9560±0.0155** | **0.9764±0.0098** | **0.9823±0.0057** |

VGGFace2 dataset, on MSCeleb-1M dataset, and on their union. Networks are learned from scratch on VGGFace2 (_scratch); Networks are first pretrained on MS1M and then fine-tuneed on VGGFace2 dataset (_ft) and the trained model can be downloaded from the Internet [1]. All experimental results are shown in Table 4 and Table 5. From the two tables we can see that for two different baseline, after using the APA method, recognition performance have been significantly improved, especially for 1:1 face verification.

### 3.5. Experiments on LFW and CPLFW Datasets

Furthermore, we evaluate our proposed method with recently reported face verification methods [2] on LFW [10] and CPLFW [31] datasets. The LFW dataset contains 13,233 web-collected images from 5749 different identities. We evaluate our methods following the standard protocol of unrestricted with labeled outside data. The Cross-Pose LFW (CPLFW) dataset is a renovation of LFW dataset [10]. It deliberately searches and selects 3,000 positive face pairs with pose difference to add pose variation to intra-class variance. Negative pairs with same gender and race are also selected to reduce the influence of attribute difference between positive/negative pairs. The CPLFW dataset is more focused on cross-pose face recognition, and is more challenging than LFW dataset. We compare our proposed method with some recent state-of-the-art methods on LFW and CPLFW datasets (Table 6). We can see that the proposed methods obtain state-of-the-art performance, achieving an accuracy of 99.68% on LFW dataset and 91.75% on CPLFW dataset.

Table 6. Verification accuracy of different methods on LFW and CPLFW.

| Method | LFW | CPLFW |
|---|---|---|
| VGG-Face [31] | 97.75% | 77.90% |
| FaceNet [23] | 99.63% | - |
| senet50_ft [31] | 99.42% | 84.45% |
| **model-B** | 99.63% | 86.92% |
| **model-E** | **99.68%** | **91.97%** |

## 4. Conclusion

In this paper, we propose a APA face alignment method to perform face recognition with images containing extreme pose variation. Our method shows that aligning all face to multi-template is better than single, frontal template, which can not only reduce intra-class variability but also correct the noise caused by alignment process. Furthermore, we also propose a simple, yet effective feature normalization method. It can be combined with the APA method to generate better feature representation of a face or template. Experiments on IJB-A and IJB-C datasets achieve state-of-the-art results for both face verification and face identification tasks.


## Acknowledgment

This work was supported by Canon Information Technology (Beijing) Co., Ltd. under Grant No. OLA18001.


[1] http://www.robots.ox.ac.uk/~vgg/data/vgg_face2/

## References


[1] Alfred M Bruckstein, Robert J Holt, Thomas S Huang, and Arun N Netravali. Optimum fiducials under weak perspective projection. *International Journal of Computer Vision*, 35(3):223–244, 1999. 3

[2] Qiong Cao, Li Shen, Weidi Xie, et al. Vggface2: A dataset for recognising faces across pose and age. In *FG*, pages 67–74. IEEE, 2018. 2, 5, 7, 8

[3] Nate Crosswhite, Jeffrey Byrne, Chris Stauffer, et al. Template adaptation for face verification and identification. *Image and Vision Computing*, 79:35–48, 2018. 5, 7

[4] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *arXiv preprint arXiv:1801.07698*, 2018. 2, 5, 7, 8

[5] Zhengming Ding, Yandong Guo, Lei Zhang, and Yun Fu. One-shot face recognition via generative learning. In *FG*, pages 1–7. IEEE, 2018. 1

[6] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012. 3

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[8] Guosheng Hu, Fei Yan, Chi-Ho Chan, et al. Face recognition using a unified 3d morphable model. In *ECCV*, pages 73–89. Springer, 2016. 1, 3

[9] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 5, 7

[10] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008. 1, 8

[11] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015. 1

[12] Brendan F Klare, Ben Klein, Emma Taborsky, et al. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *CVPR*, pages 1931–1939, 2015. 1, 7

[13] Iacopo Masi, Stephen Rawls, Gérard Medioni, and Prem Natarajan. Pose-aware face recognition in the wild. In *CVPR*, pages 4838–4846, 2016. 3, 5, 7

[14] Iacopo Masi, Anh Tu?n Tr?n, Tal Hassner, et al. Do we really need to collect millions of faces for effective face recognition? In *ECCV*, pages 579–596. Springer, 2016. 7

[15] Brianna Maze, Jocelyn Adams, James A Duncan, et al. Iarpa janus benchmark-c: Face dataset and protocol. In *ICB*, pages 158–165. IEEE, 2018. 1, 7, 8

[16] Omkar M Parkhi, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. A compact and discriminative face track descriptor. In *CVPR*, pages 1693–1700, 2014. 5

[17] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *BMVC*, volume 1, page 6, 2015. 1, 5, 6, 7, 8

[18] Rajeev Ranjan, Ankan Bansal, Hongyu Xu, et al. Crystal loss and quality pooling for unconstrained face verification and recognition. *arXiv preprint arXiv:1804.01159*, 2018. 7, 8

[19] Rajeev Ranjan, Ankan Bansal, Jingxiao Zheng, et al. A fast and accurate system for face detection, identification, and verification. *arXiv preprint arXiv:1809.07586*, 2018. 8

[20] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017. 7

[21] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D Castillo, and Rama Chellappa. An all-in-one convolutional neural network for face analysis. In *FG*, pages 17–24. IEEE, 2017. 7

[22] Swami Sankaranarayanan, Azadeh Alavi, Carlos D Castillo, and Rama Chellappa. Triplet probabilistic embedding for face verification and clustering. In *BTAS*, pages 1–8. IEEE, 2016. 7

[23] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 1, 8

[24] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014. 1, 3

[25] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, pages 1701–1708, 2014. 1, 3

[26] Paul Viola, Michael Jones, et al. Rapid object detection using a boosted cascade of simple features. In *CVPR*, volume 1, pages 511–518, 2001. 1

[27] Lin Xiong, Jayashree Karlekar, Jian Zhao, Yi Cheng, Yan Xu, Jiashi Feng, Sugiri Pranata, and Shengmei Shen. A good practice towards top performance of face recognition: Transferred deep feature fusion. *arXiv preprint arXiv:1704.00438*, 2017. 7

[28] Jiaolong Yang, Peiran Ren, Dongqing Zhang, et al. Neural aggregation network for video face recognition. In *CVPR*, pages 4362–4371, 2017. 7

[29] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 5

[30] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 3, 5

[31] Tianyue Zheng and Weihong Deng. Cross-pose lfw: A database for studying crosspose face recognition in unconstrained environments. *Beijing University of Posts and Telecommunications, Tech. Rep*, pages 18–01, 2018. 1, 8

[32] Erjin Zhou, Zhimin Cao, and Jian Sun. Gridface: Face rectification via learning local homography transformations. In *ECCV*, pages 3–19, 2018. 7