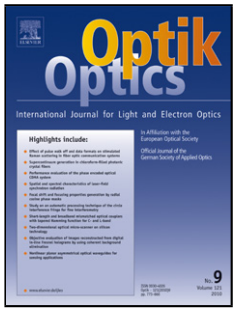
Journal Pre-proof



An End-to-End Face Recognition Method with Alignment Learning

Fenggao Tang, Xuedong Wu, Zhiyu Zhu, Zhengang Wan, Yanchao Chang, Zhaoping Du, Lili Gu

PII: S0030-4026(20)30072-3

DOI: https://doi.org/10.1016/j.ijleo.2020.164238

Reference: IJLEO 164238

To appear in: *Optik*

Received Date: 3 December 2019

Accepted Date: 16 January 2020

Please cite this article as: Tang F, Wu X, Zhu Z, Wan Z, Chang Y, Du Z, Gu L, An End-to-End Face Recognition Method with Alignment Learning, *Optik* (2020), doi: https://doi.org/10.1016/j.ijleo.2020.164238

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2020 Published by Elsevier.

Journal Pre-proof

An End-to-End Face Recognition Method with Alignment Learning

Fenggao Tang, Xuedong Wu*, Zhiyu Zhu, Zhengang Wan, Yanchao Chang, Zhaoping Du, Lili Gu
School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu 212003, China

Abstract: Many effective methods have been proposed for face recognition in the past decade and the face recognition accuracy is also gradually improved, but these algorithms usually need to perform face alignment process based on the prior knowledge of facial structure before extracting facial features. The face recognition system usually consists of face detection, face alignment, facial feature extraction, etc., which are independent of each other, and it is difficult to design and train the end-to-end face recognition model. In this paper, an end-to-end face recognition method based on spatial transformation layer is proposed. Specifically, the spatial transformation layer is placed in front of the feature extraction layer of the face recognition network, and the face region is aligned by alignment learning which requires neither prior knowledge nor artificially defined geometric transformation. The face identity category information allows the convolutional neural network to automatically learn the most appropriate face alignment. Simulation experiments on CASIA-WebFace, LFW (Labeled Face in the Wild) and YTF (Youtube Face) face database have shown that the suggested alignment learning algorithm in this paper can realize the end-to-end face recognition and can effectively improve the face recognition rate as well.

Keywords: Face recognition; Spatial transformation layer; Alignment learning; Convolutional neural network

1. Introduction

In the past decade years, the successful application of convolutional neural networks (CNN) in the image field has greatly improved the performance of computer vision tasks, such as face recognition and face verification [1, 2, 3]. The traditional face recognition methods are to construct the classification model mainly with artificially designed features, while the deep learning is to automatically learn more robust facial features through a large amount of training data. Therefore, CNN can achieve better recognition effects in the case of posture, occlusion, and illumination variation [4, 5].

However, the change of face posture is still one challenge of face recognition system in practical application. There are two ways to deal with such problems: one is to create a posture model to handle the facial posture change, Masi *et al*. [6] proposed an algorithm to

* Corresponding author. Tel.: +86 13921598973.
E-mail address: woolcn@163.com (X. Wu).

Journal Pre-proof

calculate the posture distribution of the training data and establish two CNN models which correspond to the frontal face and the profile face respectively, and an excellent face recognition effect was obtained in the case of posture change. Liao *et al*. [7] suggested a partial face recognition localization method with multi-keypoint descriptors to represent align-free faces in which the descriptors' size was determined by image content and face images. The other method is to introduce a face alignment process before facial feature extraction. Taigman *et al*. [8] developed a DeepFace network algorithm using deep learning for face recognition at the first time, and the 3D alignment method was used to solve the problem of out-of-plane rotations that traditional 2D alignment methods could not solve. Hu *et al*. [9] presented a face recognition method based on a 3D deformation model by modeling the 3D image alignment and the texture difference of the reference image, and the 3D face recognition performance was improved and had excellent robustness to posture, illumination, and occlusion changes. Recognition performance can be improved effectively by adding a face alignment step during the test phase, so a typical face recognition system is usually divided into three stages: 1) face detection; 2) facial landmarks localization and face alignment by 2D or 3D geometric transformation; 3) face recognition. In general, face alignment and facial feature extraction are performed independently. Many face alignment methods rely on accurate facial landmarks localization which is more difficult than face recognition tasks. Moreover, manual labeling of facial landmarks is much more laborious and expensive than collecting personal identity information. The geometric transformation, which usually is defined artificially, is used to complete the face alignment when the facial lanmarks are obtained. For example, a widely used method is to align the corner points of eyes, nose and mouth by similar transformation. It is not clear whether other types of 2D geometric transformations are beneficial for subsequent facial feature extraction.

However, can we assume that the location of facial landmarks doesn't need the prior knowledge for face recognition task? Since facial features can be learned through data training, can we also complete face alignment with training data? In the four stages of face recognition, other stages can be completed by data training, the prior knowledge that still needs to be manually defined in the process of correcting face images will appear to be out of place. Therefore, this paper proposes a face recognition method based on spatial transform layer, which can merge face alignment and facial feature extraction into one network framework, and any prior knowledge or artificial definition of face alignment is not required. During the training phase, the network can learn the alignment method automatically to align

Journal Pre-proof

each facial image, and then sends the aligned image to the facial feature extraction layer for further processing.

The paper is organized as follows: Several existed related works are briefly introduced in Section 2. In Section 3, we will describe the details of our model architecture. The experimental results on the LFW [10] data set and the YTF [11] data set are described in Section 4. Section 5 summarizes the work of this paper.

2. Related Work

In recent years, with the rapid development of deep learning, face recognition technology has made great progress. Traditional face recognition methods based on artificial features have not been able to meet people's needs. Since the Deepface face recognition method proposed by Taigman *et al*. [8] has shown that the deep recognition model of training large amounts of data can achieve higher recognition effect, a large number of face recognition algorithms based on deep learning technology constantly can refresh the record. Many researchers have analyzed the network structure for face recognition: He *et al*. [12] presented a deep residual network, it not only can make the neural network deeper and solve the accuracy degradation problem, but also can enhance the image feature expression ability. Hu *et al*. [13] proposed a feature recalibration method, which could automatically determine the importance of each feature channel number by learning and then could enhance useful features and suppresses unimportant features. Chen *et al*. [14] introduced mobile-net into the face recognition algorithm, although the final recognition accuracy was slightly reduced, the network running speed had been greatly improved. In addition, some researchers provided some face recognition algorithms based on the network loss function, such as Wen *et al.* [15] suggested a face recognition algorithm with the basis of Center Loss function which could increase the inter-class distance by bringing the sample closer to each category, and had excellent generalization ability for new data. Liu *et al*. [1] proposed an angular softmax loss function by adding margin to make each category having a larger decision surface, and this can obtain the effect of maximum the intra-class distance and minimum the inter-class distance. Unlike the multiplicative angular margin approach, Deng *et al*. [16] introduced a cosine additive angular margin algorithm which could make the easy training procedure and improve the face recognition rate.

In most face recognition methods based on deep learning, the input to the network is an aligned image both training and testing phases. The usual practice is to perform face alignment by 2D or 3D geometric transformation with facial landmarks. Experiments have

Journal Pre-proof

shown that correct alignment can improve face recognition performance. Parkhi *et al*. [17] found on the LFW dataset that there would be a 1% performance improvement when aligning face images. Although the 3D alignment is more accurate than the simple 2D alignment, there is no obvious advantage for the final recognition rate. Therefore, this paper mainly studies the 2D face alignment problem.

In fact, there have been many studies on the problem of geometric transformation learning, such as handwritten digit recognition and bird classification. Jaderberg *et al*. [18] proposed the network structure of the spatial transform layer, and their purpose was to learn the optimal spatial transformation mode through the network automatically and improve the robustness of the CNN to translation, scaling, rotation and even distortion of images. Because its parameters were differentiable, the spatial transform layer could train the image features through backpropagation to get the optimal transform parameters. Affected by this work, Chen *et al*. [19] improved the performance of face detection by using the spatial transformation layer. Therefore, this paper uses the spatial transformation layer to automatically learn the face alignment and integrate it with facial feature extraction to form end-to-end training.

3. Methodology

This section mainly introduces the setting of the end-to-end face recognition system and the overall structure of the system at first, and then the network settings, the loss function selection and the localization network based on the spatial transformation layer for the prediction of geometric parameters are presented, finally, the details of the spatial transformation layers of different transformation types such as identical, similarity and affine transformation are described.

3.1 Overall structure of the system

A complete face recognition system mainly includes three main parts: face detection, face alignment and face recognition. The specific process is to captures the face image or video as input through the camera and then detects it through the face detection algorithm. The aligned face is sent to the model to get identity information. Generally speaking, these three parts are designed and implemented separately. These three different computer vision tasks require different mathematical models under the traditional technology framework, so it is very difficult to integrate them together.

With the rapid development of deep learning, face detection, facial landmarks location and face recognition methods based on convolutional neural networks can achieve better

Journal Pre-proof

results. It is interesting that both of the overall network architecture and image processing are very similar. This characteristic is likely to make the design of the end-to-end face recognition system reality. The end-to-end face recognition model with satisfied people's expectation is based on the face identity and take it as the only supervised signal, and the facial model is trained in a complete end-to-end manner. However, the implementation of this method is very difficult, in order to reduce the difficulty, we use the face detection as an independent task and the face alignment and the face recognition as the main part of end-to-end design in this study.

For face detection tasks, we can use multi-tasking cascade network (MTCNN) [20] or a single stage face detector (SSH) [21] to detect faces. The MTCNN multiscales the image to form an image pyramid to detect various scales faces at first, and then classifies the faces with the first network (Pnet) and returns the coefficients of the facial borders which are rejected by non-maximum suppression. The more detailed facial borders coefficients are sent to the final network (Onet) to get the final facial borders and facial landmarks by putting the processed facial borders into the second network (Rnet) for further classification and facial borders regression. The SSH is an anchor-based face detection algorithm that uses different scale anchors to detect different size faces. The face detection effect is shown in Figure 1. In fact, any existed face detection algorithm can be applied to our system. We have proved through experiments that the impact of face detection accuracy on the final face recognition effect is negligible. The main reason for this is that the alignment network can automatically learn the appropriate alignment method to improve the accuracy and stability of the facial borders.

Figure 1. The effect of MTCNN algorithm.

For facial alignment and recognition tasks, here we design an end-to-end network which is mainly composed of a localization network that predicts the alignment parameters of facial image, sampler, and deep network for facial feature extraction. The structure diagram is shown in Figure 2. In the localization network, we use three convolutional layers whose convolution kernel size is 3*3. After each convolutional layer, we connect the batchnorm layer, the relu activation function layer and the maximum pooling layer with a kernel of size 2*2.

Journal Pre-proof

The fully connected layer is connected before the geometric parameter transformation layer, in which the projection transformation has 8 learnable parameters, the affine transformation has 6 parameters and the similar transformation has 4 parameters.

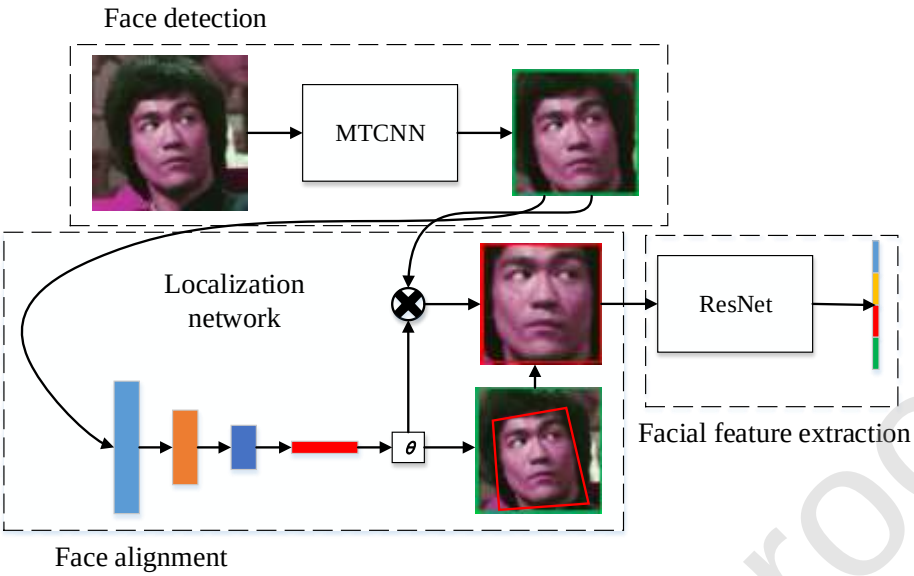


Figure 2. The overall architecture of the system.

The input image needs to be subsampled to 64x64 before entering the localization network, because it is unnecessary to use high-resolution image when calculating the transformation parameters. The residual network structure of the face feature extraction layer is shown in Table 1. Here, the first 7x7 convolution layer is replaced by a 3x3 convolution layer. Although a large convolution kernel has a larger receptive field, some details may be lost. A dropout, a full connection layer and a BN layer are used to output the face features in the last layer of the network. The cosine angle interval loss function in Arcface is used as the loss function of the network, which can not only make the network extract more discriminative features, but also accelerate the convergence speed of the model.

Table 1 Network model structure.

layers	Convolutional network structure
Conv1.x	[3×3, 64]×1, S2 [3×3, 64; 3×3, 64]×2
Conv2.x	[3×3, 128]×1, S2 [3×3, 128; 3×3, 128]×4
Conv3.x	[3×3, 256]×1, S2 [3×3, 256; 3×3, 256]×8
Conv4.x	[3×3, 512]×1, S2 [3×3, 512; 3×3, 512]×2
Dropout, FC1, BatchNorm	512

Annotation: In Table 1, Conv1.x, Conv2.x, Conv3.x, Conv4.x represent convolution units which contain

Journal Pre-proof

multiple convolution units and residual units. [3×3, 64]×1 indicates that the convolutional layer contains 1 group which contains 64 convolution kernels with size of 3×3, S2 means a step size of 2, and FC1 is a fully connected layer. [3×3, 64; 3×3, 64]×2 presents 2 groups and each convolution unit includes 64 convolution kernels with size of 3×3.

3.2 Localization Network

The essence of the localization network is used to regress the transformation parameters, and it is also a balance between the complexity of the model structure and the prediction accuracy. When the network inputs the feature image, it will output a spatial transformation parameter through a series of hidden network layers. Therefore, we hope that the network structure is as simple as possible, and the network is expected to have sufficient prediction accuracy.

In order to obtain a localization network that satisfies the requirements, we have experimented with different structures which are a combination of a convolutional layer and a fully connected layer. The experimental data set is CASIA-WebFace, its label corresponds to the facial landmarks obtained by the face detection algorithm, and the loss fuction of mean square error is utilized. We keep the same experimental conditions except the network structure. The experimental results are shown in the table 2.

Table 2 Different setting of localization network.

Setting	Network Structure	Loss (MSE)
1	1 Conv, 1FC	0.005213
2	2 Conv, 1FC	0.004795
3	3 Conv, 1FC	0.004659
4	2 Conv, 2FC	0.005169
5	3 Conv, 2FC	0.004635

It can be seen from Table 2 that the final loss value of setting 5 is the smallest, which also indicates that the network structure has lower fitting error than other structure settings, but the network parameters need to be increased compared with setting 3. However, the fitting error of setting 3 is only a little higher than setting 5, so we choose setting 3 as the localization network in this study.

3.3 Spatial transformation network

It can be concluded from [18] that the parameter transformation forms supported by the spatial transformation network include translation, scaling, affine, projection and even thin-plate spline transformation. In the traditional 2D face alignment method, the similarity transformation matrix is obtained by detecting the facial landmarks and the average face template. However, Wagner *et al*. [22] proved that the use of projection transformation in the face recognition process with large posture changes would make the model more stable and

Journal Pre-proof

could get better recognition performance. We analyze three geometric transformations including similarity transformation, affine transformation and projection transformation in this paper.

Assume that the coordinates of each pixel of the input image feature U are (x_i^s, y_i^s), and (x_i^t, y_i^t) is the coordinates of each pixel of the feature map V after being sampled and transformed by the grid generator, the space transformation function $\mathcal{T}_\theta$ is a 2D transformation function. For the projection transformation, the correspondence between (x_i^s, y_i^s) and (x_i^t, y_i^t) can be described as:

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \frac{1}{z_i^s} \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & 1 \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{1}$$

$$V_i = \sum_{h=1}^{H} \sum_{w=1}^{W} U_{wh} k(w - x_i^s, h - y_i^s) \tag{2}$$

where A_θ is composed of eight transformation parameters and z_i^s is $\theta_{31} x_i^t + \theta_{32} y_i^t + 1$. This is equivalent to convolving the image of the sampling kernel k with height H and width W, which is shown in equation (2) (where V_i represents the pixel value of the i_{th} position of the output image, and the sampling kernel $k(w - x_i^s, h - y_i^s) = max(0,1 - |w - x_i^s|) \times max(0,1 - |h - y_i^s|)$.). In the back-propagation phase, we need to calculate the gradient of the variable V_i for the eight transformation parameters. Equation (2) shows the case when $w = x_i^s$ or $h = y_i^s$, but in reality, the probability of x_i^s or y_i^s being an integer is very small. Considering that their effect on the gradient during backpropagation is negligible, we empirically set the gradient of these points to zero. For unequal points, we can caculate their gradients using chain derivation rule. Taking the parameter θ_{31} as an example, the gradient of the input coordinate points is as shown in Equations (3-5).

$$\frac{\partial V_i}{\partial \theta_{31}} = \frac{\partial V_i}{\partial z_i^s} \frac{\partial z_i^s}{\partial \theta_{31}} = \left(\frac{\partial V_i}{\partial x_i^s} \frac{\partial x_i^s}{\partial z_i^s} + \frac{\partial V_i}{\partial y_i^s} \frac{\partial y_i^s}{\partial z_i^s} \right) x_i^t = -\frac{x_i^t}{z_i^s} \left(\frac{\partial V_i}{\partial x_i^s} x_i^s + \frac{\partial V_i}{\partial y_i^s} y_i^s \right) \tag{3}$$

$$\begin{aligned} \frac{\partial V_i}{\partial x_i^s} &= \sum_{h=1}^{H} \sum_{w=1}^{W} U_{wh} \frac{\partial}{\partial x_i^s} k(w - x_i^s, h - y_i^s) \\ &= \sum_{h=1}^{H} \sum_{w=1}^{W} U_{wh} max(0,1 - |h - y_i^s|) tf(w - x_i^s) \end{aligned} \tag{4}$$

$$tf(w - x_i^s) = \begin{cases} 0, |w - x_i^s| > 1 \\ 1, 0 < w - x_i^s \leq 1 \\ -1, -1 \leq w - x_i^s \leq 0 \end{cases} \tag{5}$$

For similarity transformations, we can define as:

Journal Pre-proof

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = A_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \lambda\cos\alpha & -\lambda\sin\alpha & t_1 \\ \lambda\sin\alpha & \lambda\cos\alpha & t_2 \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \quad (6)$$

where α is the angle of rotation, λ is the scale factor, t_1 and t_2 are the horizontal displacement and the vertical displacement. Similarly, the gradient of V_i for λ and α is shown in Equations (7) and (8).

$$\frac{\partial V_i}{\partial \alpha} = \frac{\partial V_i}{\partial x_i^s}\frac{\partial x_i^s}{\partial \alpha} + \frac{\partial V_i}{\partial y_i^s}\frac{\partial y_i^s}{\partial \alpha} = \frac{\partial V_i}{\partial x_i^s}(t_2 - y_i^s) + \frac{\partial V_i}{\partial y_i^s}(x_i^s - t_1) \quad (7)$$

$$\frac{\partial V_i}{\partial \lambda} = \frac{\partial V_i}{\partial x_i^s}\frac{\partial x_i^s}{\partial \lambda} + \frac{\partial V_i}{\partial y_i^s}\frac{\partial y_i^s}{\partial \lambda}$$

$$= \frac{\partial V_i}{\partial x_i^s}(x_i^t\cos\alpha - y_i^t\sin\alpha) + \frac{\partial V_i}{\partial x_i^s}(x_i^t\sin\alpha + y_i^t\cos\alpha) \quad (8)$$

As mentioned in the introduction, the widely used face recognition alignment scheme is a non-reflective similarity transformation. However, it is unclear how different types of 2D conversions can affect face recognition performance. In order to explore the type of transformation that is most suitable for face recognition, we will train four different models including identical, similarity, affine and projection, transformations under the condition that keeping the training set and the rest of the network structure unchang. For the identical transformation, the detected face region is directly cropped at the center for recognition. Corresponding results and face accuracy with LFW and YTF are presented in Section 4.

4 Experimental results and analysis

In order to verify the effectiveness of the end-to-end face recognition method, CASIA-WebFace [23] is used as the training set. Although increasing the training set size or multi-patch feature fusion can improve the performance of face recognition effectively, the main work of this study is to verify the feasibility of end-to-end learning and the impact of face alignment on different geometric transformation types on recognition results. Therefore, we train different types of alignment networks on CASIA-WebFace and only use a single patch feature for identification, and then test the performance of the model on the LFW and YTF datasets.

Although increasing the training set size or multi-patch feature fusion can effectively improve the performance of face recognition, the main work of this study is to verify the feasibility of end-to-end learning and the impact of face alignment on the recognition results of different geometric transformation types. Therefore, we train different types of alignment networks on CASIA-WebFace using only one single patch feature for identification and then

Journal Pre-proof

test the performance of the model on LFW and YTF datasets.

The CASIA-WebFace dataset contains 10575 people with total 494,414 face images, in which everyone has a number of pictures ranging from tens to hundreds, and we use horizontal flipping for data augmentation. For each picture, the face detection algorithm described in the Section 3 is used to detect the face, and then the face image is cropped using an area slightly larger than the detection frame. The cropped image is directly sent to the end-to-end face recognition network. and the fixed-size area is directly cropped as the training data by the image center point if the face is not detected.

Assuming that the batch size is 64, the loss function is Arcface and its cosine angle interval parameter is 0.5 in our study. The optimizer uses the random gradient descent method with momentum (the momentum value and the initial network learning rate are set to be 0.9 and 0.01). After each 10000 iterations attenuation, we find that we can obtain the best results when the learning rate of alignment network is 10-100 times smaller than that of the recognition network. Maybe the main reason is that the loss value generated by the recognition network is much larger than the value of the transformation parameter.

We test the trained models on two widely used unconstrained face recognition benchmark datasets named LFW and YTF. The LFW face dataset contains 5749 people including total 13233 face images. The YTF dataset contains 3425 video sequences with 1595 different identities and an average of 2.15 videos for each person. We need to verify 5000 pairs of videos (2500 pairs of videos from the same person, 2500 pairs of videos from different people), and the final recognition rate can be obtained by averaging these 5000 similarity values calculated from each pair of videos. Both datasets are allowed to use 10 folds cross-validation. We use the sum of corresponding positions between each tested image and its mirror image feature as the final feature representation and use the cosine distance to calculate the similarity between these two images. The optimal classification threshold is determined by training 9 datasets, and then the remaining datasets is tested.

Table 3. Face verification performance on LFW and YTF datasets.

Method	trainset	LFW	YTF
FaceNet	200M	99.76%	95.1%
DeepID2+	0.2M	99.47%	93.2%
Center Face[15]	0.7M	99.28%	94.9%
A-softmax[1]	0.46M	99.47%	95.0%
AM-Softmax[3]	0.46M	99.58%	96.2%
Ours (Identical)	0.46M	98.01%	93.4%
Ours (Similarity)	0.46M	98.87%	94.2%
Ours (Affine)	0.46M	99.02%	94.6%

Journal Pre-proof

Ours (Projective)	0.46M	99.24%	94.6%

Table 3 shows the numerical results of the verification performance, and several observations can be found from the verification accuracy. At first, among the four types of transformations, the identity transformation results in the lowest verification accuracy (98.01% and 93.4%) which is consistent with previous research, and the explicit alignment of face images can significantly improve the efficiency of face recognition. Secondly, projection transformation (99.24%) is more suitable for face recognition than similar transformation (98.87%) and affine transformation (99.02%). This is not surprising because the projection transformation can more accurately describe the imaging process of most camera.

Figures 3 and 4 show the corresponding ROC curves which represent the relationship between the false positive rate and the true positive rate of the sample at different classification thresholds, and this can reflect the quality of face recognition model. We can see that the verification accuracy of face recognition models with different geometric transformation types on the YTF data set is similar comparing with the LFW data set, in which the curves representing similar transformation and affine transformation almost coincide with each other. This is expected because the average face features are extracted from a series of video sequence images when testing on YTF and it can greatly alleviate the impact of face pose transformation. However, face alignment is still helpful for the generalization of face recognition model.

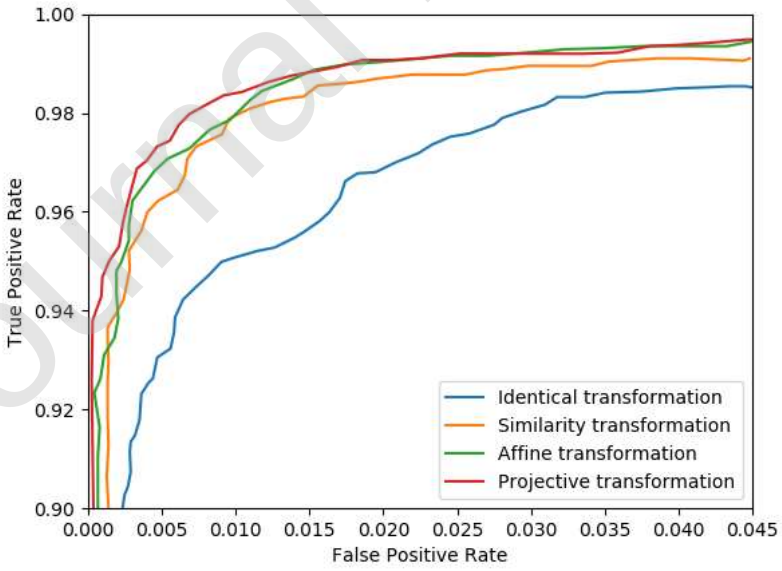


Figure 3. ROC curve for face verification on LFW.

Journal Pre-proof

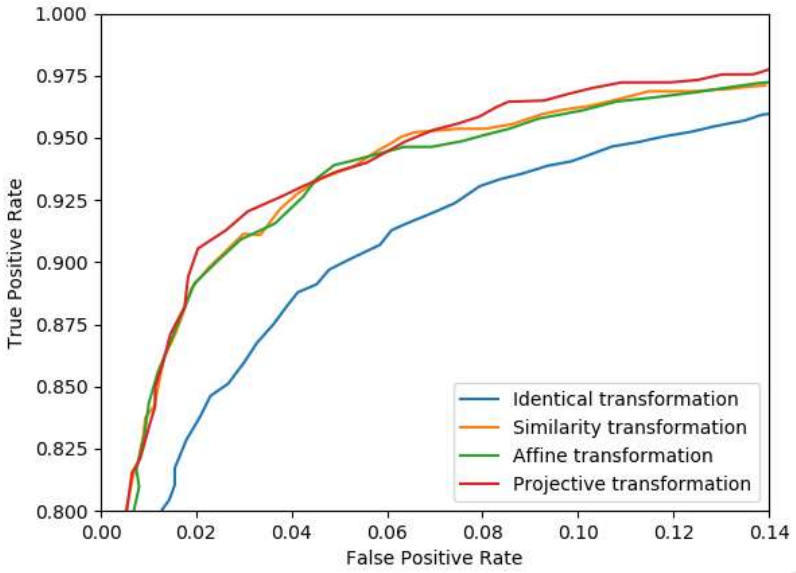


Figure 4. ROC curve for face verification on YTF.

5. Conclusions

We propose an end-to-end trainable framework in which the face alignment and the facial feature extraction can be jointly trained using only the personal identity as the supervising signal. Therefore, explicit knowledge about human face characteristics and artificially defined geometric transformation principles are no longer needed for face alignment in the recognition task. In fact, our proposed method provides a foundation for the future implementation of end-to-end face recognition system, and this can be easily extended to other fine-grained image recognition tasks. The other future work is to use more training data and more carefully designed data enhancement strategies to improve the robustness of transformation prediction for extreme posture changes and exaggerated facial expressions.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 61671222) and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX19_1693).

Journal Pre-proof

References

[1] W. Liu, Y. Wen, and Z. Yu. Sphereface: deep hypersphere embedding for face recognition [C]. IEEE Conference on Computer Vision and Pattern Recognition, 2017: 6738-6746.

[2] H. Wang, Y. Wang, and Z. Zhou. CosFace: large margin cosine loss for deep face recognition [C]. IEEE Conference on Computer Vision and Pattern Recognition, 2018: 2456-2469.

[3] F. Wang, J. Cheng, and W. Liu. Additive margin softmax for face verification [J]. IEEE Signal Processing Letters, 2018, 45(67): 657-669.

[4] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: a unified embedding for face recognition and clustering [C]. IEEE Conference on Computer Vision and Pattern Recognition, 2015: 815-823.

[5] C. Ding, and D. Tao. A comprehensive survey on pose-invariant face recognition [J]. Acm Transactions on Intelligent Systems and Technology, 2015, 7(3): 37-46.

[6] I. Masi, S. Rawls, and G. Medioni. Pose-aware face recognition in the wild [C]. Conference on Computer Vision and Pattern Recognition, 2016: 4838-4846.

[7] S. Liao, A. K. Jain, and S. Z. Li. Partial face recognition: Alignment-free approach [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(5): 1193-1205.

[8] Y. Taigman, M. Yang, and M. Ranzato, Deepface: closing the gap to human-level performance in face verification [C]. IEEE Conference on Computer Vision and Pattern Recognition, 2014: 1701-1708.

[9] G. Hu, F. Yan, and C.H. Chan. Face recognition using a unified 3D morphable model [C]. European Conference on Computer Vision, 2016, pp. 73-89.

[10] G.B. Huang, M. Ramesh, and T. Berg. Label faces in the wild: a database for studying face recognition in unconstrained environments [C]. Technical Report University of Massachusetts Amherst, 2007: 07-49.

[11] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched back-ground similarity [C]. IEEE Conference on Computer Vision and Pattern Recognition, 2011: 529-534.

[12] K. He, X. Zhang, and S. Ren. Deep residual learning for image recognition [C]. IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770-778.

[13] J. Hu, L. Shen, and S. Albanie. Squeeze-and-excitation networks [C]. IEEE Conference on Computer Vision and Pattern Recognition, 2017: 4324-4335.

[14] S. Chen, Y. Liu, and X. Gao. Mobilefacenets: efficient cnns for accurate real-time face verification on mobile devices [J]. Biometric Recognition, 2018: 428-438.

[15] Y. Wen, K. Zhang, and Z. Li. A discriminative feature learning approach for deep face recognition [C]. European Conference on Computer Vision, 2016: 499-515.

[16] J. Deng, J. Guo, and S. Zafeiriou. Arcface: additive angular margin loss for deep face recognition [C]. arXiv preprint arXiv:1801.07698, 2018.

[17] O.M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition [C]. The British Machine Vision Conference, 2015: 6-18.

[18] M. Jaderberg, K. Simonyan, and A. Zisserman. Spatial transformer networks [C]. Conference and Workshop on Neural Information Processing Systems, 2015: 2017-2025.

[19] D. Chen, G. Hua, and F. Wen. Supervised transformer network for efficient face detection [J]. European Conference on Computer Vision, 2016: 122-138.

[20] K. Zhang, Z. Zhang, and Z. Li. Joint face detection and alignment using multitask cascaded convolutional networks [J]. IEEE Signal Processing Letters, 2016, 23(10): 1499-1503.

[21] M. Najibi, P. Samangouei, and R. Chellappa. SSH: single stage headless face detector [C]. IEEE International Conference on Computer Vision, 2017: 567-578.

[22] A. Wagner, J. Wright, and A. Ganesh. Toward a practical face recognition system: robust alignment and

Journal Pre-proof

illumination by sparse representation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(2): 372-386.

[23] D. Yi, Z. Lei, and S. Liao. Learning face representation from scratch [J]. arXiv preprint arXiv: 1411.7923, 2014.

Journal Pre-proof