

Recursive Spatial Transformer (ReST) for Alignment-Free Face Recognition

Wanglong Wu^{1,2} Meina Kan^{1,3} Xin Liu^{1,2} Yi Yang⁴ Shiguang Shan^{1,3} Xilin Chen¹

¹Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing 100190, China

²University of Chinese Academy of Sciences, Beijing 100049, China

³CAS Center for Excellence in Brain Science and Intelligence Technology

⁴Huawei Technologies Co.,Ltd., Beijing 100085, China

{wanglong.wu, meina.kan, xin.liu, shiguang.shan, xilin.chen}@vip.ict.ac.cn yangyi16@huawei.com

Abstract

Convolutional Neural Network (CNN) has led to significant progress in face recognition. Currently most CNN-based face recognition methods follow a two-step pipeline, i.e. a detected face is first aligned to a canonical one pre-defined by a mean face shape, and then it is fed into a CNN to extract features for recognition. The alignment step transforms all faces to the same shape, which can cause loss of geometrical information which is helpful in distinguishing different subjects. Moreover, it is hard to define a single optimal shape for the following recognition, since faces have large diversity in facial features, e.g. poses, illumination, etc. To be free from the above problems with an independent alignment step, we introduce a Recursive Spatial Transformer (ReST) module into CNN, allowing face alignment to be jointly learned with face recognition in an end-to-end fashion. The designed ReST has an intrinsic recursive structure and is capable of progressively aligning faces to a canonical one, even those with large variations. To model non-rigid transformation, multiple ReST modules are organized in a hierarchical structure to account for different parts of faces. Overall, the proposed ReST can handle large face variations and non-rigid transformation, and is end-to-end learnable and adaptive to input, making it an effective alignment-free face recognition solution. Extensive experiments are performed on LFW and YTF datasets, and the proposed ReST outperforms those two-step methods, demonstrating its effectiveness.

1. Introduction

Face recognition is one of the most important applications of computer vision. The task is to recognize or identify a given subject based on the face appearance. As with many other vision tasks, face recognition has benefited from the

Figure 1. Conventional face recognition pipeline

powerful deep learning models, particularly Convolutional Neural Network (CNN), and has met prominent boost of recognition accuracy [16, 14, 17]. Though equipped with new models, most CNN-based face recognition approaches still follow the conventional recognition pipeline. As illustrated in Fig.1, given an input image, face detection is first performed to obtain the bounding box of each face. Then the detected faces are aligned to a canonical one for more robust feature extraction, and finally the aligned faces are used to recognize the subjects.

In the step of face alignment, detected faces are transformed to a canonical one according to the affine relationships between their facial landmarks and a pre-defined mean shape. Since inferring the affine transformation is trivial, the main task of alignment is to predict the position of facial landmarks (i.e. face shape) with the face appearance as input. Typical face alignment methods include Active Shape Model (ASM), Active Appearance Model (AAM), which employ Principal Component Analysis (PCA) to build statistical models of face shape and appearance. More recently, there have been methods exploiting deep neural networks to model the regression from face appearance to face shape [15, 23], and they can achieve high prediction accuracy even in the presence of large variations due to pose, expression, etc.

Taking aligned faces as input, many approaches have been designed for face recognition. Previous works mainly employ hand-crafted features, e.g. Gabor [11] and Local Binary Patterns (LBP) [1], and linear models, as in the classic Eigenfaces [19] and Fisherfaces [2]. In recent research, deep learning methods have shown great superiority over

previous ones by using learned representations and highly non-linear models [17, 16, 14]. The pioneering DeepFace [17] adopts CNN with locally connected layers, and it uses a 3D alignment model to transform all detected faces to the frontal view before they are fed into the network. The DeepID series [16] learn an ensemble of neural networks with different local face patches aligned in different ways, which provides a rich set of features for face verification. FaceNet [14] directly learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. PAMs [12] learns Pose-Aware Models (PAM) for frontal, half-profile and full-profile faces respectively to handle pose variations.

As a prerequisite step for face recognition in most cases, face alignment helps improve the robustness of recognition, but it also brings issues which lie in two folds. First, after faces are aligned to the canonical one, the positions of landmarks on all face images are almost identical. While variations between faces including those in rotation, pose and expression, which are less irrelevant to subject identification, are eliminated, some identification-relevant characteristics are also weakened, e.g. the geometrical structure formed by facial landmarks. Second, though the prior alignment aims to assist recognition by removing identification-irrelevant variations, alignment and recognition are optimized according to different objectives when treated separately as two independent steps. Therefore, the transformation of aligning to a pre-defined mean shape may be suboptimal for the final recognition. Moreover, it is hard to define a single optimal shape for face recognition in all scenarios, since people have great diversity in facial characteristics.

One way to deal with the problems caused by the two-step pipeline is to directly discard the alignment step and perform face recognition with unaligned detected faces, as is done in FaceNet [14]. In order to cover as many face variations as possible and build a robust recognition system without alignment, FaceNet uses a tremendous amount of data for training, i.e. 100M-200M face images consisting of about 8M different identities. Considering the expensive cost of dealing with such big data, this solution is not economical and hardly favorable.

The spatial transformer [7] is a learnable module which explicitly allows spatial manipulations, e.g. affine transformation, of data within CNNs. The Spatial Transformer Network (ST-Net) can automatically learn an optimal transformation for classification task and has achieved state-of-the-art performance on digital number recognition. But since spatial transformer can only model linear affine transformations, it is still hard for ST-Net to learn complex variations, especially non-rigid ones.

Inspired by spatial transformer, we design a novel Recursive Spatial Transformer (ReST) module for CNN which allows face alignment and recognition to be jointly opti-

Figure 2. Examples of aligned faces in each recursion of ReST.

mized in one network in an end-to-end fashion. ReST is able to model complex transformations in a progressive way. As illustrated in Fig.2 and Fig.3, given a detected face, ReST transforms it recursively to make it more adequate for recognition. In each recursion, a further transformation is generated and performed based on the previously transformed face to get one step closer to the optimal one. In this way, large variations can be handled progressively, thus improving the robustness of the model to large variations among faces. Furthermore, to model non-linear, i.e. non-rigid, variations, the proposed ReST is extended to a hierarchical form (HiReST) and multiple ReST modules are used to account for different local face regions. With the whole face divided into different local regions in different granularity, the variations in each local region are approximately rigid and thus can be modeled as an affine transformation by a single ReST. During learning, both the transformation parameters and the hierarchical structure are automatically optimized according to the recognition objective, leading to more adequate alignment and more robust recognition model. By using ReST, the face recognition process becomes alignment-free and the problems with the conventional two-step pipeline are naturally resolved. To demonstrate the effectiveness of the proposed ReST for face recognition, extensive experiments are performed on the Labeled Faces in the Wild dataset (LFW) [5] and YouTube Faces dataset (YTF) [20]. And our recognition model with ReST outperforms those two-step methods.

The rest of the paper is organized as follows. Section 2 describes the ReST module and the extended hierarchical form. Section 3 and 4 presents experimental analysis and evaluations. Section 5 concludes this paper.

2. Method

As mentioned, in the typical two-step methods, a face is first aligned via an affine transformation determined from the automatically or manually labeled facial landmarks, and then fed into a DCNN to recognize it. By contrast, our proposed DCNN with Recursive Spatial Transformation attempts to jointly optimize the face alignment and classifica-

Figure 3. Overview of the proposed ReST integrated in a CNN for alignment-free face recognition.

tion within one network.

As shown in Fig.3, our approach consists of two parts, the Recursive Spatial Transformation (ReST) followed by the DCNN classification. Given a detected face, the ReST endeavors to optimize an affine transformation that can transform the detected face to a new one based on which the following DCNN can achieve better classification objective. The ReST and the DCNN is optimized together with one classification objective (such as the softmax loss) in an end-to-end scheme, therefore the aligned face achieved from the ReST can well match the succeeding DCNN classification and is also adaptive for each input face promising better performance. The ReST is organized in a recursive structure attempting to characterize the variations progressively for better alignment. Besides, this ReST module can be integrated into any convolutional architecture.

Hereinafter, for clear presentation, we use boldface uppercase, boldface lowercase, and lowercase letters to denote a matrix (e.g. \mathbf{A}) or a function, a vector (e.g. \mathbf{a}), and a scalar (e.g. a) respectively.

2.1. DCNN with ReST

The most important part of our proposed method is the ReST as the succeeding DCNN can be any kind of off-the-shelf convolutional architecture. The goal of ReST is to optimize a spatial transformation to align the input face image. The ReST follows the recursive structure including three parts: Convolution layers \mathbf{C} , Localization network \mathbf{F} and Spatial Transformation layer \mathbf{T} .

Given an input image $\mathbf{X} \in \mathbb{R}^{w \times h}$ with width as w and height as h , the convolution feature maps achieved as $\mathbf{C}(\mathbf{X})$ through the convolutions layers are used for better representation as they are usually more informative. Here, the number of convolution layers can be one or more depending on the difficulty of the problem. Furthermore, the localization layer predicts the spatial transformation parameter $\mathbb{R}^{2 \times 3}$ by taking the feature maps $\mathbf{C}(\mathbf{X})$ as input:

$$\mathbf{t}_i = \mathbf{F}(\mathbf{C}(\mathbf{X}_i)). \quad (1)$$

The localization layer \mathbf{F} is usually designed as fully connected layer for regression. Here \mathbf{t}_i is a 6-dimensional parameters of 2D affine transformation including rotation, scale and translation as below:

$$\mathbf{t}_i = \begin{bmatrix} t_{i1} & t_{i2} & t_{i3} \\ t_{i4} & t_{i5} & t_{i6} \end{bmatrix} \quad (2)$$

Finally, the spatial transformation layer produces the transformed feature map by sampling from the input \mathbf{X} according to the spatial transformation parameter \mathbf{t}_i , formulated as $\mathbf{T}(\mathbf{X}, \mathbf{t}_i)$. The new face image can be further transformed by feeding it into this pipeline again, forming a recursive structure ReST. For the i^{th} ($i = 1, 2, \dots, K$) recursion, we have:

$$\mathbf{X}_i = \mathbf{T}(\mathbf{X}_{i-1}, \mathbf{t}_{i-1}), \quad (3)$$

$$\mathbf{t}_i = \mathbf{F}(\mathbf{C}(\mathbf{X}_i)) \quad (4)$$

where $\mathbf{X}_0 = \mathbf{X}$, $\mathbf{t}_0 = \mathbf{I}$, \mathbf{I} is the identity matrix, and K is the maximum number of recursion. For image \mathbf{X}_{i-1} , the spatial transformation \mathbf{T} with parameter \mathbf{t}_{i-1} is computed as follows:

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_{i-1} & y_{i-1} & 1 \end{bmatrix} \mathbf{t}_{i-1} \quad (5)$$

where (x_i, y_i) is the position index of a pixel in \mathbf{X}_i .

With the transformed face image \mathbf{X}_K from the last recursion, any kind of convolutional neural network structure can be exploited, formulating an end-to-end learning framework integrating both the alignment and recognition. For example, \mathbf{X}_K can be followed by an AlexNet [9] with softmax loss. In this whole network, the ReST and AlexNet are optimized together under the same softmax loss, so the ReST can align the input face image \mathbf{X} into a new one which is best suitable for the following DCNN promising better classification performance. The derivatives of feedforward and feedbackward of the ST are the same as that in [7], and please refer to [7] for more details. An illustration of the recursively transformed face image is shown in Fig.2.

2.2. DCNN with Hierarchical ReST

As we know, the face variations are generally non-rigid and facial deformation caused by pose and expression can not fully modeled by a linear affine transformation in ReST. To well characterize the complex non-rigid variations, following the divide and conquer strategy we further designs

Figure 4. The structure of HiReST-3

Figure 5. The structure of HiReST-9

a hierarchical ReST as shown in Fig.5, referred as HiReST. In HiReST, the whole face is divided into several hierarchical regions, and each region is equipped with a ReST which is used to determine rigid area and its corresponding spatial transformation parameters. As observed in the experiments, the preceding ReSTs characterize those major affine transformation, while the succeeding ReSTs characterize those non-rigid transformation by modeling part of it as rigid affine transformation within a smaller region.

We take the AlexNet as an exemplar DCNN architecture to illustrate two DCNN with hierarchical ReST used in this work, HiReST-9 in Fig.5 and HiReST-3 in Fig.4. HiReST-9 is a typical hierarchical structure with three regions in each layer and two layers in total. Straightforwardly, the second layer can directly take the aligned image from the first layer as input, but generally the second layer can also take the convolution feature maps of the aligned image as input such as that shown in Fig.5. HiReST-3 is a degraded version where only one layer with three regions, and this version is suitable for those scenarios with moderate variations. For both HiReSTs, each ReST in the hierarchical structure is initialized with an empirically defined region as shown in the figures, and after that the whole end-to-end network will automatically adjust the region (including the position, scale, rotation etc.) for each ReST according to the objective of the whole network to promise an optimal region and affine transformation. An illustration of the recursively transformed face image is shown in Fig.6.

Compared with the baseline AlexNet, HiReST-9 and HiReST-3 have more complex structure, however the computational complexity of the network excluding the ReST, i.e., just the DCNN part, only a little bit increases as shown in Table. 1. In the next section, we will describe a speed up method for ReST, named as Fast ReST. With the fast ReST, the computational complexity of our whole network and that of the AlexNet are roughly comparable. In other words, our proposed DCNN with Hierarchical ReST can achieve better performance efficiently.

Figure 6. Examples of aligned faces in each recursion of Hierarchical ReST.

2.3. Fast ReST

To make the ReST more efficient, we speed up it from two aspects. Firstly, we share several convolution layers between the ReST and the following DCNN. For example, in ReST the first two convolution layers of the DCNN are shared with the ReST as the first two convolution layers are more representative than discriminative so they can be used as the common layer of both ReST and DCNN.

Secondly, we observe that the first several recursions usually characterize those major transformation, such as rotation. For these major transformation, convolution feature maps from the transformed image are similar as directly transforming the convolution feature maps, i.e.,

$$C(T(X_i, i)) \approx T(C(X_i), i) \quad (6)$$

Therefore, in the first several recursions, e.g. k ($1 < k < K$), the spatial transformation is applied on the convolution feature maps rather than the face image to avoid the time-consuming convolution operations as follows:

$$X_i = \begin{cases} T(X, i), & i = k + 1 \\ T(X_{i-1}, i-1), & i > k + 1 \end{cases} \quad (7)$$

$$i = \begin{cases} F(T(C(X), i-1)), & 0 < i < k+1 \\ 0 & 0 & 1 \\ F(C(X_i)), & K > i > k+1 \end{cases} \quad (8)$$

Briefly speaking, in the first k recursions, the convolution is computed only once, i.e. $C(X)$, and the spatial transformation is directly applied on convolution feature maps $C(X)$, and the spatial transformation parameters are accumulated together which is applied on the input image X after the k recursion. In the succeeding ($> k$) recursions, the spatial transformation achieved in each recursion is directly applied on the transformed image from previous recursion without accumulation since that the spatial transformations in the late recursions are usually small so the convolution feature maps from the transformed image are different from that achieved by directly transforming the convolution feature maps.

Table 1. Details of **HiReST**. conv[N,w,s, p] denotes a convolution layer which has N filters of size w w, with stride s and p pixel padding; ReST[t,s] is a ReST with recursive times t and s denotes whether to do a average pooling after ReST; FC[N] is a fully connected layer with N units; and max[w,s] is a w w max-pooling with stride s. NUM is the number of face identities on the training set. All the convolution layers followed by a batch normalization [6] layer. The ReST in stage 1 of HiReST shared convolution layers each other and the ReST in stage 3 have no convolution layers. The comp.1 is the theoretical time complexity [4] excluding the ReST compared to AlexNet and comp.2 is the theoretical time complexity when taking ReST in account.

Networks	AlexNet	ReST	HiReST-3	HiReST-9
input	112*112*3			
stage 1	conv[96,7,3,2]	ReST[4,0] conv[96,7,3,2]	3*ReST[4,0] 3*conv[64,7,3,2]	3*ReST[4,0] 3*conv[64,7,3,2]
stage 2	max[3,2] conv[192,5,2,1]	max[3,2] conv[192,5,2,1]	3*max[3,2] 3*conv[96,5,2,1]	3*max[3,2] 3*conv[96,5,2,1]
stage 3	max[3,2] conv[384,3,1,1] conv[384,3,1,1] conv[256,3,1,1]	max[3,2] conv[384,3,1,1] conv[384,3,1,1] conv[256,3,1,1]	3*max[3,2] 3*conv[192,3,1,1] 3*conv[192,3,1,1] 3*conv[128,3,1,1] concat[3]	3*[max[3,2] + 2*ReST[2, 1]] 9*conv[128,3,1,1] 9*conv[128,3,1,1] 9*conv[96,3,1,1] concat[9]
stage 4	max[3,2] fc[4096] fc[2048] fc[NUM] softmax	max[3,2] fc[4096] fc[2048] fc[NUM] softmax	max[3,2] fc[4096] fc[2048] fc[NUM] softmax	max[3,2] fc[4096] fc[2048] fc[NUM] softmax
comp.1	1	1	0.86	1.09
comp.2	1	1.42	1.01	1.23

The above speeded up ReST from two aspects are referred as Fast ReST in this work. In most CNN architectures, the first several convolution layers account for the major time complexity, and in our Fast ReST the first several convolution layers are shared between the ReST and DCNN, and also only few even no recursion (when $k = K - 1$) need to re-compute these convolution feature maps. Therefore, those DCNN architectures with the Fast ReST only take very limited additional time computation compared with the DCNN without ReST.

2.4. Discussion

Advantages of ReST. The proposed end-to-end learning approach of DCNN with ReST has several advantages: 1) It is an end-to-end learning framework, so the face alignment is optimized to be most suitable for the following DCNN classification. 2) The recursive structure of ReST makes the detected face been aligned progressively, meaning an easier task in each recursion, and thus obtains a better alignment. 3) The hierarchically structured ReST disperses the non-rigid transformation into multiple rigid transformations leading to more accurate alignment. 4) The ReST is a general module, and several existing methods can be reformulated in this framework: when the depth of recursion is 0, i.e. no recursion, the proposed ReST method degenerates to a typical CNN, such as AlexNet; when the depth of recursion is 1, the proposed method is a kind of DCNN equipped

with the so-called spatial transformer layer [7]; generally the depth of recursion can be larger than 1 for better performance.

Differences with the ST [7]. 1) The ST can be considered as a special case of our ReST when the recursion depth is 1. Our recursive ST is more general with better non-linearity for large transformations. 2) Besides, multi-stream ST in [7] is organized in parallel structure while ours is organized in hierarchical structure which is more flexible for complex non-rigid transformation.

Differences with the typical two-step methods. 1) In the typical two-step methods, the face alignment and the face recognition is conducted separately and with different objective, so the face alignment is not necessarily optimal for the following recognition. In contrast, in our ReST, the face alignment is learnable and jointly optimized with the DCNN classification under the same objective, and thus the optimized face alignment can well match with the classification leading to better performance. 2) In the face alignment step of the two-step methods, usually all faces are aligned to a pre-defined mean shape, which means that the geometry information between the landmarks is lost after alignment which actually reduces the distinguishability of different subjects. On the contrary, in our ReST, the affine transformation of each face is adaptively determined by the ReST which can preserve those beneficial geometry.

3. Experiments

In this section, we will investigate the proposed DCNN with ReST by evaluating the performance w.r.t. different architectures and comparing with the state-of-the-art methods on two wild challenging datasets.

3.1. Datasets and Experimental Settings

In all experiments, three wild face datasets are used, i.e., CASIA-WebFace [21], Labeled Faces in the Wild database(LFW) [5] and YouTube Faces(YTF) dataset [20]. Among them, the CASIA-WebFace is used for training, LFW and YTF are used for testing.

The **CASIA-WebFace** dataset [21] is a large-scale dataset containing about 10,575 subjects and 500,000 images from Internet. This unconstrained dataset has accelerated the development of face recognition in the wild.

The **LFW** dataset [5] is a large dataset collected in the unconstrained environment for evaluating face verification in the wild, which has 13,233 images from 5,749 individuals. On this dataset, we follow the standard evaluation protocol of unrestricted with labeled outside data, i.e., training on the outside labeled CASIA-WebFace, and testing on the 10 folds verification set in view 2 of LFW. For this face verification evaluation, the performance is reported in terms of the mean accuracy (mAC) according to the standard protocol. Furthermore, Best Rowden et al. develop a face identification protocol on this LFW dataset [3], including close-set face identification measured by rank-1 recognition rate and open-set face identification measured by the Detection and Identification Rate (DIR) as well as False Alarm Rate (FAR). This identification protocol is also used for the evaluation. More details about the standard protocol can be found in [5] and [3].

The **YTF** dataset [20] is a large unconstrained video dataset which collects 3,425 YouTube videos of 1,595 subjects(a subset of the celebrities in the LFW). All the videos were downloaded from YouTube. On the average, there are 2.15 videos for each subject and the length of each video clip is about 181 frames. The standard protocol is similar as that on LFW, i.e., face video verification, which consists of 5,000 video pairs organized into 10 splits. On this dataset, the performance is also reported in terms of the mean accuracy (mAC) according to the standard protocol [20].

For all three datasets, we employ the SURF Cascade [10] to do face detection, and then resize the detected face into 128×128 . For those methods that do not need face alignment, the detected faces are directly used, while for those that need prerequisite face alignment, we use the CFAN [23] to detect five landmarks (2 eye centers, 1 nose tip, and 2 mouth corners) and then align the face images via the affine transformation according to the detected facial landmarks.

In all experiments, for our proposed ReST, the DCNN follows the AlexNet architecture [9]. Three different ar-

Table 2. Face verification and open-set face identification of the proposed ReST on LFW dataset w.r.t. different numbers of recursions.

#Recursions	mAC(%)	DIR(%)@ FAR=1%	DIR(%) FAR=5%
0	97.37 \pm 0.13	53.3	48.3
1	98.03 \pm 0.26	60.4	53.0
2	98.08 \pm 0.22	61.3	55.1
3	98.25 \pm 0.21	64.5	59.4
4	98.38 \pm 0.15	65.4	60.9
5	98.20 \pm 0.28	64.0	58.7

chitectures are investigated, i.e. ReST, HiReST-3, and HiReST-9, and the detailed network structures are shown in Table 1. Fast-ReST is employed for all architectures with $k = K$ as illustrated in Sec. 2.3. All models are trained on CASIA-WebFace and tested on LFW and YTF. Source code will be available along with this work.

3.2. Investigation of ReST w.r.t. Different Settings

3.2.1 Depth of Recursion for ReST.

One of the most important parameters in our ReST is the number of recursion as it determines the fitting ability of the ReST. Here, we evaluate the performance of ReST w.r.t. the number of recursion on the LFW dataset. The ReST is equipped with AlexNet without hierarchical structure, as shown in Fig.3. The results are shown in Table.2. When the number of recursion is 0, i.e., the DCNN with no recursive ReST, the network actually degenerates to the AlexNet itself. When with only one recursion, our ReST degenerates to the ST-Net [7]. The proposed ReST with more than one recursions outperforms it with no or only one recursions, demonstrating the effectiveness of our recursive spatial transformations. Besides, the performance increases when with more recursions benefited from the progressively aligning the detected face promising a better transformed face for the following DCNN, but a little degenerates when with too many recursions as only very little variations are left to characterize which is hard to formulate a reliable transformation. In the following experiments, the number of recursions is set as 4 for all ReSTs.

3.2.2 HiReST for Alignment-free Face Recognition.

In the HiReST, the hierarchical structure plays an important role as it determines the grain of the rigid transformations. Here, we compare three different hierarchical structures, ReST (i.e., no hierarchy), HiReST-3 (1-layered hierarchy), and HiReST-9 (2-layered hierarchy). In the HiReST, there are two factors that could affect the performance, i.e. the hierarchy and the recursive ST. For comprehensive comparison, we also evaluate the DCNN with

Table 3. Face verification and open-set face identification of the proposed HiReST on LFW dataset w.r.t. different hierarchical structures with and without alignment respectively.

Method	With Alignment (0 recursion)			Alignment-free (4 recursions)		
	mAC(%)	DIR(%)@ FAR=1%	DIR(%)@ FAR=5%	mAC(%)	DIR(%)@ FAR=1%	DIR(%)@ FAR=5%
AlexNet	98.06	55.2	48.6	97.37	53.3	48.3
ReST	98.06	55.2	48.6	98.38	65.4	60.9
HiReST-3	98.45	62.1	56.2	98.62	71.1	66.5
HiReST-9	98.70	63.2	58.2	98.90	77.2	72.8

Table 4. The comparisons of face verification on LFW and YTF.

Method	LFW	YTF	#Nets	#Train Images	Train Align.	Test Align.
DeepFace [17]	97.35	91.4	3	4M	Yes	Yes
DeepID2+ [16]	99.47	93.2	200	0.3M	Yes	Yes
FaceNet [14]+Align.	99.63	95.1	1	200M	No	Yes
VGG [13]	97.27	91.6	1	2.6M	No	Yes
VGG [13]+Embed.	98.95	97.3	1	2.6M	No	Yes
FaceNet [14]	98.87	-	1	200M	No	No
HiReST-9+(Ours)	99.03	95.4	1	0.5M	No	No

the same hierarchy structure but without ReST, i.e. setting the recursion number as 0 based on the pre-aligned face images. When with no recursion the ReST degenerates to the AlexNet, meaning the same performance. All results are shown in Table 3. From the comparisons, we can see that the proposed HiReST which models the face alignment and recognition together in an end-to-end framework performs much better than those two-steps methods with the same architecture, with an improvement even up to 14% under the open-set identification protocol with HiReST architecture. On the more challenging IJB-A [8] dataset, HiReST-9 achieves significant improvement on both verification and recognition tasks, up to 18.3%(TAR@FAR=0.001), 6.4%(TAR@FAR=0.01), and 2.6%(Rank-1) compared with AlexNet. Besides, the HiReST-9 performs the best demonstrating the effectiveness of the proposed hierarchical spatial transformation module as it can flexibly characterize those non-rigid transformations. These comparisons also show that the proposed ReST or HiReST is an effective end-to-end framework for alignment-free face recognition.

4. Comparison with Existing Methods

Furthermore, the proposed ReST is compared with the state-of-the-art methods, including DeepFace [17], DeepID2+ [16], FaceNet [14], VGG [13], COTS [3], and WST Fusion [18] on both LFW and YTF datasets. As most of these Deep networks are much deeper than the AlexNet, we modify the AlexNet to a deeper one, by replacing the

5×5 filters in HiReST-9 with two 3×3 filters and inserting a 1×1 convolution layer before each 3×3 convolution layer following the work [22]. This deeper structure of HiReST-9 is denoted as HiReST-9+, and it has almost the same complexity as HiReST-9.

The comparisons for face verification on LFW and YTF are shown in Table 4. Among these methods, only FaceNet and our HiReST are fully alignment-free, i.e. both training and testing data are not pre-aligned, and our method outperforms FaceNet even with much less training data. Besides, our method outperforms DeepFace and VGG, and is comparable to DeepID2+ which ensembles about 200 networks.

Moreover, the proposed method is also compared with the state-of-the-art methods for face identification on LFW as shown in Table 5. Our method achieves the best per-

Table 5. The performance of closed-set and open-set face identification on LFW.

Method	Rank-1(%)	DIR(%)@ FAR=1%	#Nets
COTS-s1 [3]	56.7	25	1
COTS-s1+s4 [3]	66.5	35	2
DeepFace [17]	64.9	44.5	3
WST Fusion [18]	82.5	61.9	1
DeepID2+ [16]	95.0	80.7	200
HiReST-9+(Ours)	93.4	80.9	1

formance of open-set face identification, and in the experiments of closed-set face identification, our method outperforms COTS, DeepFace and WST Fusion. As observed, our method is only a little worse than DeepID2+, but our method uses only one network and is fully alignment-free.

As seen from these comparisons, our proposed ReST method achieves quite promising performance for the totally alignment-free face recognition, demonstrating the effectiveness of the proposed.

5. Conclusions and Future Works

In this work, we propose a Recursive Spatial Transformer (ReST) module in the DCNN architecture forming an end-to-end learning framework which can jointly optimize the face alignment and face recognition under the same objective for alignment-free face recognition. The proposed ReST is recursive, learnable, and adaptive for each input face image. Moreover, it is applicable for non-rigid transformation by being designed hierarchically, and thus make the HiReST an effective alignment-free face recognition model. As evaluated on several datasets, the proposed HiReST outperforms those two-steps methods. In future, we will endeavor to design non-linear transformation rather than affine transformation in the end-to-end learning framework to better characterize those quite challenging variations.

6. Acknowledgements

This work was partially supported by 973 Program under contract No. 2015CB351802, Natural Science Foundation of China under contracts Nos. 61390511, 61650202, 61402443, 61272321, and the Strategic Priority Research Program of the CAS (Grant XDB02070004).

References

- [1] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *European Conference on Computer Vision (ECCV)*, pages 469–481. 2004.
- [2] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 19(7):711–720, 1997.
- [3] L. Best-Rowden, H. Han, C. Otto, B. F. Klare, and A. K. Jain. Unconstrained face recognition: Identifying a person of interest from a media collection. *IEEE Transactions on Information Forensics and Security (TIFS)*, 9(12):2144–2157, 2014.
- [4] K. He and J. Sun. Convolutional neural networks at constrained time cost. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5353–5360, 2015.
- [5] 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.
- [6] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015.
- [7] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2008–2016, 2015.
- [8] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1931–1939, 2015.
- [9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.
- [10] J. Li, T. Wang, and Y. Zhang. Face detection using surf cascade. In *IEEE International Conference on Computer Vision Workshops (ICCVw)*, pages 2183–2190, 2011.
- [11] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image processing (TIP)*, 11(4):467–476, 2002.
- [12] I. Masi, S. Rawls, G. Medioni, and P. Natarajan. Pose-aware face recognition in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4838–4846, 2016.
- [13] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference (BMVC)*, 2015.
- [14] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.
- [15] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3476–3483, 2013.
- [16] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2892–2900, 2015.
- [17] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708, 2014.
- [18] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Web-scale training for face identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2746–2754, 2015.
- [19] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–591, 1991.

- [20] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 529–534, 2011.
- [21] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. CoRR, abs/1411.7923, 2014.
- [22] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In European Conference on Computer Vision (ECCV), pages 818–833. 2014.
- [23] J. Zhang, M. Kan, S. Shan, and X. Chen. Leveraging datasets with varying annotations for face alignment via deep regression network. In IEEE International Conference on Computer Vision (ICCV), pages 3801–3809, 2015.