Towards End-to-End Face Recognition Method through Spatial Transformer

Yanfang Wu1,2, Xiaobo Lu1,2,3 and Chen Qin1,2

1 School of Automation, Southeast University, Nanjing 210096, China
2 Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing, 210096, China
3 Corresponding author: xblu2013@126.com

Abstract. Face recognition technology is widely applied in daily life, but in most methods, similarity or affine transformation is employed to align face images according to five facial landmarks. The face alignment module is implemented independently, thus it’s difficult in end-to-end training. In this paper, the main purpose is to design a towards end-to-end trainable face recognition method based on indoor scenes. Due to that spatial transformer can implement any parametrizable transformation, we joint it with recognition network, making end-to-end training possible. Simultaneously, any prior knowledge on facial landmarks isn’t required. The model jointly with spatial transformer can achieve 0.3% higher accuracy than similarity transformation. Most downsampling methods ignore the sampling theorem, making convolutional networks not shift-invariant. We replace max-pooling by MaxBlurPool in spatial transformer network, and the accuracy is improved by 0.25%.

1. Introduction

Nowadays, face recognition technology is more and more widely applied in our life. In public security, there are some typical applications, such as intelligent access control and driver's license verification. In economy, it’s significant for bank card identity verification. It’s also applied for the attendance system in company and school. The majority of the applications have something in common, it’s that cameras are placed at a fixed position. Therefore, it’s necessary to recognize pose-variant faces. For pose-variant face recognition, face alignment is vital.

The key of a face recognition framework can be summarized as the face detection, face alignment and face recognition. Deep learning is widely applied in face detection and recognition, and there are a great many achievements. In terms of face detection, there are MTCNN [1], SSH [2], DSFD [3], RetinaFace [4] and so on. It has also made great progress in face recognition, mainly in network structure and loss. In terms of network structure, there are Vggface [5], ResNet [6] and so on. In terms of loss, there are Center loss [7], Range loss [8], SphereFace [9], CosFace [10], and ArcFace [11]. For face alignment, it’s performed by fitting a 2D or 3D geometric transformation between detected facial landmarks and certain predefined landmarks. 3D transformation does not show significant improvements over simple 2D alignments in recognition accuracy [12]. Thus in most face recognition methods, 2D transformation is widely used, especially similarity transformation based on five facial landmarks (eye centers, nose, and mouth corners). This causes that it’s difficult in designing and training end-to-end face recognition models. With the proposition of spatial transformer network [13], end-to-end training is possible.

In this paper, since we assume that the face recognition method is applied in indoor scene, lighting variance has little effect on recognition. It’s quite enough that the camera manages to capture enough
high resolution images. In order to build a towards end-to-end trainable face recognition framework, we joint spatial transformer network and recognition network, optimized with ArcFace. Furthermore, conventional downsampling methods ignore the sampling theorem, on this account, we replace max-pooling by MaxBlurPool [14] in spatial transformer network. To verify the effectiveness of the method, we conduct a series of experiments. It demonstrates that spatial transformer and MaxBlurPool contribute to improve recognition accuracy, and end-to-end training is feasible.

The rest of the paper is organized as follows. In next section, we describe the details of our face recognition method. In section 3, implementation details and experimental results are presented. The conclusion is followed in section 4.

2. Approach

2.1. Overall Architecture

Our face recognition method includes three major parts, which are detection, alignment and recognition. For face alignment, the five facial landmarks are usually adopted to perform similarity transformation. But in this paper, we employ spatial transformer network to align face crops and facial landmarks don't need to be obtained, thus the overall framework can be end-to-end trainable. For building a fully end-to-end face recognition framework, the face detector can actually serve as a region proposal network or an attention model to propose candidate facial regions, so that it can be easily jointed to the alignment and recognition network [15]. We just emphasize the end-to-end trainable network that joints spatial transformer network and recognition network, as shown in Figure 1.

![Figure 1. The overall architecture of the face recognition method](image)

2.2. Face Detection Network

As the most widely used face detector, MTCNN not only has excellent performance in frontal face detection, but also has a high accuracy for pose-variant face. Compared with RetinaFace, MTCNN has higher detection speed simultaneously with good detection accuracy. Thus, we adopt it to predict the face location, actually, any off-the-shelf face detector can be probably used in our method.

According to the bounding boxes detected by MTCNN, face regions are cropped. Specifically, we use a slightly enlarged version of the bounding box to include the whole head (factor is 0.3). Then the face crops are resized to $112 \times 112$ pixels, which is used as the input of spatial transformer network.

2.3. Spatial Transformer Network

Any parametrizable transformation can be implemented by spatial transformer, such as similarity, affine, projective and so on [13]. A spatial transformer network consists of a localization network and a spatial transformer which can be split into a grid generator and a sampler). The localization network predicts the 2D transformation parameters based on the detected face image. The spatial transformer warps the face image according to the predicted transformation parameters to produce the aligned face. Conventional downsampling methods like max-pooling, strided-convolution or average-pooling ignore the sampling theorem, resulting in convolution networks not shift-invariant. For better performance, we replace max-pooling by MaxBlurPool in spatial transformer network. MaxBlurPool not only works for anti-aliasing but also has increased accuracy in ImageNet classification.
2.3.1. Localization network. The localization network we adopt has three convolution layers and two fully connected layers, inspired by [15]. The kernel sizes of the three convolutional layers with 24, 48, 96 nodes are 5 × 5, 3 × 3, and 3 × 3 respectively. The output size of the two fully connected layers are both 64. PReLU and MaxBlurPool are used after each convolution layer. Specially, MaxBlurPool combines blur filter and subsampling. The first step of max-pooling is to get the max value in slide window (stride 2), then conduct sampling. While MaxBlurPool has a low-pass filter before sampling, moreover, stride and padding are 1 when getting the max value. After the last layer, there is the regression of the geometric transformation parameters $\theta$, and affine transformation is employed here.

\[
\begin{align*}
\begin{pmatrix} x'_{i,1} \\ x'_{i,2} \\ x'_{i,3} \end{pmatrix} &= \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{pmatrix} \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ x_{i,3} \end{pmatrix} \\
&= \begin{pmatrix} \theta_{11} \theta_{21} \theta_{31} x_{i,1} + \theta_{12} \theta_{22} \theta_{32} x_{i,2} + \theta_{13} \theta_{23} \theta_{33} x_{i,3} \\ \theta_{11} \theta_{21} \theta_{31} y_{i,1} + \theta_{12} \theta_{22} \theta_{32} y_{i,2} + \theta_{13} \theta_{23} \theta_{33} y_{i,3} \\ \theta_{11} \theta_{21} \theta_{31} z_{i,1} + \theta_{12} \theta_{22} \theta_{32} z_{i,2} + \theta_{13} \theta_{23} \theta_{33} z_{i,3} \end{pmatrix}
\end{align*}
\]

Where $i$ refers to the $i_{th}$ target point in aligned face. The sizes of aligned face and the input can be different, and the aligned face size is 112×112 here.

Although the source coordinates are calculated, the input image can’t be sampled directly on account that the generated source coordinates may not be integers. Image interpolation method is needed, generally, it’s a bilinear sampling kernel. The process of sampling can be expressed as

\[
V_i = \sum_{h} \sum_{w} U_{wh} \max(0,1-|x_{i}^j - w|) \max(0,1-|y_{i}^j - h|)
\]

Where $U_{wh}$ is the pixel value at location $(w,h)$ in the input, $V_i$ is the value of pixel $i$ in the output.

2.3.2. Spatial transformer. According to the affine transformation parameters $\theta$ (2×3) predicted by the localization network and the target coordinates $(x_i', y_i')$ in the output image, the grid generator generates its source coordinates $(x_i, y_i)$ in the input image. The transformation can be expressed as

\[
\begin{align*}
\begin{pmatrix} x_{i,1} \\ x_{i,2} \\ x_{i,3} \end{pmatrix} &= \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{pmatrix} \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ x_{i,3} \end{pmatrix} \\
&= \begin{pmatrix} \theta_{11} \theta_{21} \theta_{31} x_{i,1} + \theta_{12} \theta_{22} \theta_{32} x_{i,2} + \theta_{13} \theta_{23} \theta_{33} x_{i,3} \\ \theta_{11} \theta_{21} \theta_{31} y_{i,1} + \theta_{12} \theta_{22} \theta_{32} y_{i,2} + \theta_{13} \theta_{23} \theta_{33} y_{i,3} \\ \theta_{11} \theta_{21} \theta_{31} z_{i,1} + \theta_{12} \theta_{22} \theta_{32} z_{i,2} + \theta_{13} \theta_{23} \theta_{33} z_{i,3} \end{pmatrix}
\end{align*}
\]

\[V_i = \sum_{h} \sum_{w} U_{wh} \max(0,1-|x_{i}^j - w|) \max(0,1-|y_{i}^j - h|)
\]

Where $U_{wh}$ is the pixel value at location $(w,h)$ in the input, $V_i$ is the value of pixel $i$ in the output.

2.4. Face Recognition Network

Generally, different face recognition methods are mainly different in training data, network structure, and loss. For instance, ResNet and Inception-ResNet can obtain better performance than VGG network and Inception v1 [11]. Thus, considering the high generalization capability of ResNet demonstrated in various visual recognition problems, we employ ResNet-36 (given in [9]) as the basic network for recognition feature extraction. In addition, we add batch normalization after every convolution layer, and activation is PReLU. To check how the embedding setting affects the model performance, Deng et al. [11] proposed five different options in last several layers (Option-A ~ Option-E). Since the option E (BN-Dropout-FC-BN) can obtain the best performance, we use it after the last convolutional layer. The ResNet-36 produces a 512 dimensional output feature vector to capture the face feature embedding.

3. Experiments

In order to demonstrate the effectiveness of spatial transformer network and MaxBlurPool, four sets of experiments are carried based on Resnet36E, namely ResNet36E-Arc(Pre-aligned), Resnet36E-Arc, ST-Resnet36E-Arc, STBlur-Resnet36E-Arc, while keeping the training set and training parameters unchanged except the first experiment. The training samples of ResNet36E-Arc(Pre-aligned) are
aligned by MTCNN (the most common method), and the training parameters are same as other experiments. The difference of the network architectures is whether the net includes spatial transformer network and MaxBlurPool in the experiments.

3.1. Implementation Details

We employ a clean version of CASIA [16] and our own dataset as our training data. Our dataset includes 10k photos of 200 unique individuals, and the face images of every individual have five views (frontal, left side, right side, bottom, and vertical). Before training, we need to do preprocessing for the train set. We use MTCNN for face detection, when there’re more than one face in the image, we select to pick the face closed to the center. The longer side of the detected face bounding box is selected as its height and width, then the bounding boxes are extended by a factor 0.3 to include the whole head. Finally, the face crops are resized to $112 \times 112$ pixels, which is used as network input.

Especially, for ResNet36E-Arc(Pre-aligned), the training samples are aligned by MTCNN and the detected bounding boxes are not extended. To increase the diversity of data and further improve the generalization ability of the network, some data augmentation operations are employed, such as mirroring the images, changing image contrast, doing color shift along RGB axes and doing smooth filtering. All augmentation operations are applied with 50% probability. Each pixel ($[0, 255]$) in RGB images is normalized by subtracting 127.5 and then dividing by 128.

During training, we set batch size to 32. In order to increase the practical batch size, iteration size is set to a certain value. We set momentum to 0.9 and weight decay to 5e-4. The dropout ratio is set as 0.4 in Option-E. Before training ST-Resnet36E-Arc, to make it convergence easier, we trained a ST-Resnet36E model jointly optimized with softmax loss and center loss as initial weights. The coefficient of the center loss is set to 0.008. The learning rate of the recognition net (lr_rec) is 0.02 and is divided by 10 at 15K iterations, while the learning rate of the localization net (lr_loc) is 100 times smaller than lr_rec. The optimizer is SGD, and the training is finished at 20K iterations. Then train ST-Resnet36E with ArcFace, the feature scale s and the angular margin m are set as 64 and 0.5 respectively. The lr_loc begins with 0.01 and decays by 0.7 every 10K iterations, while lr_rec is 10K times smaller than lr_rec. The training is finished at 40K iterations. During training STBlur-Resnet36E and Resnet36E, the training settings and steps are kept to be the same. When training ResNet36E(Pre-aligned), it also keeps the same training steps and settings except training set.

3.2. Experimental Results

In the verification experiments, we have collected 30 persons’ face images, and do the same preprocessing like training set, which are saved as the pre-established database. The samples collected from every participant include five views like our own training dataset. Noticeably, the side view refers to the face rotated by an angle less than 90 degrees relative to the frontal face. After establishing the database, it’s to capture the images containing the participants by a camera real-time for testing. We extract the deep features from the output of the FC1 layer. The feature and its flipped feature of each captured image are concatenated as the feature representation for recognition. The similarity score between the corresponding features of a pair of images (the image from pre-established database and the image captured by camera) is computed by the cosine distance. The similarity threshold is set to 0.6. The recognition results of STBlur-Resnet36E for the images captured by camera are shown as Figure 3, and the red marks denote the predicted name and the similarity.

Table 1 and Table 2 show the face verification performance of the four sets of experiments, and in Table 1, the contrast of Resnet36E and ST-Resnet36E indicates that face alignment is vital for face recognition. Moreover, from Table 1, it’s demonstrated that spatial transformer network and MaxBlurPool contribute to improve the recognition accuracy (improvements of about 0.3% and 0.25%). Correspondingly, omission ratio and fallout ratio are reduced. Table 2 visually illustrates that the accuracy of frontal face recognition is bigger than that of other views. And the accuracy of the four views are increased with the employment of spatial transformer network and MaxBlurPool.
Table 1. Face verification performance

| Network             | Accuracy | Omission Ratio | Fallout Ratio |
|---------------------|----------|----------------|---------------|
| Resnet36E(Pre-aligned) | 88.45%   | 4.72%          | 6.83%         |
| Resnet36E           | 84.17%   | 6.89%          | 8.94%         |
| ST-Resnet36E        | 88.75%   | 4.65%          | 6.60%         |
| STBlur-Resnet36E    | 89.05%   | 4.54%          | 6.41%         |

Table 2. Face verification performance on different view

| Network             | Frontal | Side    | Bottom  | Vertical |
|---------------------|---------|---------|---------|----------|
| Resnet36E(Pre-aligned) | 94.57%  | 89.05%  | 85.48%  | 84.70%   |
| Resnet36E           | 88.79%  | 85.92%  | 81.44%  | 80.53%   |
| ST-Resnet36E        | 94.88%  | 89.18%  | 85.93%  | 85.01%   |
| STBlur-Resnet36E    | 95.15%  | 89.58%  | 86.21%  | 85.26%   |

Figure 3. (a) is the person image from database, (b) ~ (f) are the five views of the person
4. Conclusion
In this paper, we design a towards end-to-end trainable face recognition method based on indoor scenes, and the input is the image captured real-timely by a camera. In the method, there’re three networks, which are face detection network, spatial transformer network, and face recognition network. Spatial transformer network makes that face alignment and facial feature extraction can be jointly trained and any prior knowledge on facial landmarks isn’t required. In addition, we replace max-pooling by MaxBlurPool in spatial transformer network. The experimental results show that the model has a good performance with less variable factors like lighting in indoor scene. And it’s demonstrated that spatial transformer network and MaxBlurPool contribute to improve recognition accuracy.

5. Acknowledgments
This work was supported by the National Natural Science Foundation of China (No.61871123), Key Research and Development Program in Jiangsu Province (No.BE2016739) and a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

6. References
[1] Zhang K, Zhang Z, Li Z and Qiao Y 2016 Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks IEEE Signal Processing Letters 23 1499-1503.
[2] M. Najibi, P. Samangouei, R. Chellappa and L. Davis 2017 SSH: Single Stage Headless Face Detector IEEE Int. Conf. on Computer Vision (Venice, ITALY) pp 4885-94.
[3] Li J, Wang Y, Wang C, Tai Y, Qian J, Yang J, Wang CJ, Li JL and Huang F 2018 DSFD: Dual Shot Face Detector arXiv:1810.10220 [cs.CV].
[4] Deng J, Guo J, Zhou Y, Yu J, Kotsia I and Zafeiriou S 2019 RetinaFace: Single-stage Dense Face Localisation in the Wild arXiv:1905.00641 [cs.CV].
[5] Parkhi O.M., Vedaldi A. and Zisserman A 2015 Deep face recognition Proc. of the British Machine Vision Conf.
[6] He K, Zhang X, Ren S, and Sun J 2016 Deep Residual Learning for Image Recognition IEEE Conf.on Computer Vision and pattern recognition (Las Vegas) pp 770-8.
[7] Wen Y, Zhang K, Li Z and Qiao Y 2016 A Discriminative Feature Learning Approach for Deep Face Recognition European Conf.on Computer Vision (Amsterdam) pp 499-515.
[8] Zhang X, Fang Z, Wen Y, Li Z and Qiao Y 2017 Range Loss for Deep Face Recognition with Long-Tailed Training Data IEEE Int. Conf.on Computer Vision (Venice, ITALY) pp 5419-28.
[9] Liu W, Wen Y, Yu Z, Li M and Raj B 2017 SphereFace: Deep Hypersphere Embedding for Face Recognition IEEE Conf.on Computer Vision and pattern recognition (Honolulu, HI) pp 6738-46.
[10] Wang H, Wang Y, Zhou Z, Ji X, Gong D, Zhou J, Li Z and Liu W 2018 CosFace: Large Margin Cosine Loss for Deep Face Recognition IEEE Conf.on Computer Vision and pattern recognition (Salt Lake City) pp 5265-74.
[11] Deng J, Guo J, Xue N and Zafeiriou S 2018 ArcFace: Additive Angular Margin Loss for Deep Face Recognition arXiv:1801.07698 [cs.CV].
[12] Banerjee S, Brogan J, Krizaj J, Bharati A, RichardWebster B, Struc V, Flynn P and Scheirer, W 2016 To Frontalize or Not To Frontalize: Do We Really Need Elaborate Pre-processing To Improve Face Recognition? IEEE Winter Conf.on Applications of Computer Vision (NV/CA) pp 20-29.
[13] Jaderberg M, Simonyan K, Zisserman A and Kavukcuoglu K 2015 Spatial Transformer Networks Annual Conf.on Neural Information Processing Systems (Montreal, CANADA) 28.
[14] Richard Zhang 2019 Making Convolutional Networks Shift-Invariant Again Int. Conf. on Machine Learning (Long Beach) pp 7324-34.
[15] Zhong Y, Chen J and Huang B 2017 Toward End-to-End Face Recognition Through Alignment Learning IEEE Signal Processing Letters 24 1213-17.
[16] Yi D, Lei Z, Liao S and Li S 2014 Learning Face Representation from Scratch arXiv:1411.7923 [cs.CV].