·**Article**·

End-to-end spatial transform face detection and recognition

Hongxin ZHANG*, Liying CHI

CAD&CG State Key Lab, Zhejiang University, Hangzhou 310058, *China*

* **Corresponding author,** zhx@cad.zju.edu.cn

Received: 19 November 2019 **Accepted:** 30 March 2020

Citation: Hongxin ZHANG, Liying CHI. End-to-end spatial transform face detection and recognition. Virtual Reality & Intelligent Hardware, 2020 2(2): 119—131

DOI: 10.1016/j.vrih.2020.04.002

Abstract **Background** Several face detection and recognition methods have been proposed in the past decades that have excellent performance. The conventional face recognition pipeline comprises the following: (1) face detection, (2) face alignment, (3) feature extraction, and (4) similarity, which are independent of each other. The separate facial analysis stages lead to redundant model calculations, and are difficult for use in end-to-end training. **Methods** In this paper, we propose a novel end-to-end trainable convolutional network framework for face detection and recognition, in which a geometric transformation matrix is directly learned to align the faces rather than predicting the facial landmarks. In the training stage, our single CNN model is supervised only by face bounding boxes and personal identities, which are publicly available from WIDER FACE and CASIA-WebFace datasets. Our model is tested on Face Detection Dataset and Benchmark (FDDB) and Labeled Face in the Wild (LFW) datasets. **Results** The results show 89.24% recall for face detection tasks and 98.63% accuracy for face recognition tasks.

Keywords Face detection; Face recognition; Spatial transform; Feature fusion

1 Introduction

Due to the rapid development of VR and AR applications, face detection and recognition have garnered wide attention. Face detection plays an important role in computer vision and pattern recognition due to its fundamental facial analysis stages, including face recognition, age/gender recognition, emotion recognition, and face transformation. Several face detection methods have been proposed since Viola and Jones proposed a real-time object detection framework[1]. However, face detection encounters many challenges, such as illumination, pose, rotation, and occlusion.

In the past, these challenges were solved by combining different models or using different hand-crafted features. In recent years, the convolutional neural network (CNN) has been used to achieve higher performance in computer vision tasks[2,3].

Common face recognition methods use faces with known identities to train classification models and consider the intermediate layer as the feature expression. However, human faces are not always frontal; hence, it is important to extract spatially invariant features from face patches, using largepose transformation. Almost all the methods use a facial landmark predictor[4–9] to locate the position of facial landmarks. Subsequently, they perform facial alignment by fitting the geometric transformation between the predicted facial landmarks and the pre-defined landmarks.

The common pipeline for face recognition comprises the following: (1) face detection, (2) face alignment, (3) feature extraction, and (4) similarity calculation, which are independent of each other. Several methods[10–12] have focused on how to efficiently extract features from face patches to make the in-class closer and outclass farther in the feature space, using different loss functions[13,14].

However, separate facial analysis stages lead to redundant model calculations and are difficult for use in end-to-end training. Since joint learning can boost the performance of individual tasks, such as joint face detection and alignment[6], many multi-task methods for facial analysis have been proposed[6,15,16].

In this paper, a novel end-to-end trainable convolutional network framework for face detection and recognition is proposed. The proposed framework benefits from the strong object detection ability of the Faster R-CNN[17] framework. In the proposed framework, the facial landmark prediction and alignment stages are replaced by a spatial transformer network (STN)[18], in which a geometric transformation matrix is obtained to align the faces rather than predicting the facial landmarks. Compared with facial landmark prediction network, STN is smaller and more flexible for almost any feature, which makes the network end-to-end trainable. Moreover, face detection and recognition tasks can share common lower features to reduce unnecessary feature calculation. This end-to-end network improves the performance and is easy to extend to multi-task problems.

This paper makes the following contributions:

(1) A novel end-to-end trainable convolutional network framework is proposed for face detection and recognition. In the framework, STN is used for face alignment. It is trainable and does not need to be supervised by labeled facial landmarks.

(2) In the proposed framework, the detection part, recognition part, and STN share common low-level features, which makes the model smaller and reduces unnecessary calculations.

(3) A single CNN model is supervised only by face bounding boxes and personal identities, which are publicly available from WIDER FACE[19] and CASIA-WebFace[20] datasets. When the model was tested on Face Detection Dataset and Benchmark (FDDB)[21] and Labeled Face in the Wild (LFW)[22] datasets, it had 89.24% recall on the FDDB dataset and 98.63% accuracy on the CASIA dataset.

This paper is organized as follows. Section 2 presents related works. Section 3 describes the proposed framework. Section 4 presents the experimental results. Section 5 concludes the paper with a brief discussion and future works.

2 Related work

Facial detection has rapidly developed in recent decades. In 2001, Viola and Jones proposed a cascaded Adaboost framework using Haar-like features for real-time facedetection[1]. In recent years, CNN has shown its powerful ability in computer vision and pattern recognition. Many CNN-based object detection methods have been proposed[2,3,17,23–26]. Ren et al. improved the region-proposal-based CNN method and proposed Faster RCNN framework[17]. The Faster RCNN framework introduced anchors method and made region proposal a CNN classification problem that could be trained in the whole net during the training stage. The end-to-end trainable Faster R-CNN network is faster and more powerful, achieving 73% mAP in the VOC2007 dataset with VGGnet. Other studies[27–29] have used Faster R-CNN framework to solve various face detection problems and achieved promising results.

2.1 Face recognition

Face recognition tasks take the aligned face as the input. The aligned face is obtained after a similarity transformation, in which a similarity matrix is obtained by mapping the facial landmarks to a pre-defined landmark template. After reducing the facial pose difference, especially roll rotation (rotation around Z-axis), a universal representation is then obtained from the normalized faces. Huang et al. showed that performing similarity transformation before extracting facial representation could have a 1% accuracy improvement on the LFW dataset[22]. FaceNet splits face image into several patches, feeds them into independent networks, and integratesthe features as final representation. However, other methods feed the full image into a single delicate network. The former considers differentareas of the face, while the latter focus on better network design and stricter loss function: classification-based loss and metric-based loss. Furthermore, hard mining and balanced data distribution improve face recognition performance.

2.2 Joint learning

The pipeline of face recognition system sequentially combines face detection, face alignment, and face recognition, as individual parts. This is suboptimal, as these three parts depend on one another but cannot optimize together. Multi-task learning has shown improvements in natural language processes[30,31], neural machine translation, attribute classification[32,33], person re-id[33], segmentation, etc. Without ignoring the link of the tasks, joint learning can take advantage of the inter-relationships and make the tasks benefit from each other. Chen et al. first performed a study on joint learning for face detection and alignment[34]. Zhang et al. combined face classification and landmark regression as cascaded single shallow networks[6]. Deng et al. manually annotated five landmarks on WILDER FACE dataset and trained a one-shot RetinaNet face detector, which output box and landmark together, an unsupervised dense regression branch is used for improving performance[35]. Ranjan et al. proposed HyperFaceto simultaneously process face detection, facial landmark localization, head pose estimation, and gender recognition, except face recognition[15]. Furthermore, Ranjan et al. proposed an all-in-one network that processes different facial tasks, including face recognition[16]. However, the input faces were cropped and aligned with the original image. Thus, the all-in-one network cannot perform face detection tasks efficiently.

2.3 Spatial transformer network

Jaderberg et al. introduced a new learnable module, spatial transformer, that explicitly allows the spatial manipulation of data within a network[18]. The spatial transformer directly learns the best geometric transformation of the input feature maps and transforms the features to make them robust for rotation, scale, and translation. It directly performs regression on the transformation parameters and is only supervised by the final recognition objectives. It is widely used in text detection and recognition[36,37], person re-id[38], and human pose estimation[39]. Inspired by STN, Zhong et al. proposed a single recognition model that takes unaligned face as input and replaces normalization by STN[40]. Furthermore, Wu et al. proposed a recursive spatial transformer for more accurate face recognition[41].

In this paper, we combine face detection and recognition tasks into a single network. As STN is a play-and-plug module and back propagates the gradient, it is a bridge between the detection and recognition parts. The recognition part takes the aligned feature rather than the aligned face image as input. The two parts share common low-level features both in training and test strategies, which is important for training and reducing multiply-add flops.

3 Methodology

A novel end-to-end trainable convolutional network framework for face detection and recognition is proposed. In the framework, a geometric transformation matrix was obtained by STN to align the faces, rather than predicting the facial landmarks. Compared with facial landmark prediction network, STN is smaller and more flexible for almost any feature. This makes the network end-to-end trainable such that face detection and recognition tasks could share common low-level features and reduce unnecessary feature calculations. The detection part in the proposed architecture is based on Faster R-CNN framework, which is a state-of-the-art object detector. Meanwhile, the recognition part is based on a simplified Residual Network (ResNet)[43].

3.1 Architecture

Typically, a face recognition system requires cropped face patches as input, which are the results of a pre-trained face detector. A deep CNN then processes the cropped faces and obtains distinguishable features for the recognition task. The common pipeline of face recognition task includes the following stages: (1) face detection, (2) face alignment, (3) feature extraction, and (4) similarity calculation. However, the separate stages lead to redundant model calculations and are difficult for use in end-to-end training. In this work, face detection and face recognition tasks shared common low-level features. Using STN in face alignment, the model became end-to-end trainable because the gradient was calculated and backward computation was possible.

In this section, the whole architecture is described into three parts. Section 3.2 describes the detection part. Section 3.3 introduces the STN, while Section 3.4 describes the recognition part.

3.2 Detection part

We used VGG-16 network[42], based on Faster R-CNN framework[29], as the pre-trained model for face detection. The network was pre-trained on ImageNet[44] for image classification so that we could benefit from its various image representations for different object categories. Region proposal network (RPN), a fully convolutional network, was used to predict the bounding boxes and the objectless scores for the anchors, using the convolution results from the VGG network. The anchors were defined with different scales and ratios to obtain the translation invariance. The RPN then outputs a set of proposal regions that were most likely to be the targets.

In the original Faster R-CNN framework, the ROI Pooling operator pools the features of the proposal regions extracted from the convolutional feature maps into a fixed size. The fully-connected layers then predict the offset of the bounding boxes and the scores of the proposal regions. In the proposed method, STN[45]is applied to transform the region features between the ROI Pooling layer and the fully-connected layers. STN is used to obtain a transformation matrix that makes the input features spatially invariant. It is flexible such that it can be adopted in any convolutional layer without obvious costs in training and testing. In the detection network, STN shares the features of the proposal regions to regress the transform matrix parameters, and a transform operation uses the predicted parameters to transform the input features. The transformed output features are then passed for classification and regression.

3.3 Spatial transformer network

Although CNN achieves good performance in feature extraction,it is not effective to be spatially invariant for the input data. For facial analysis problems, the input faces could be obtained in a variety of different conditions, such as different scales and rotations. To solve this problem, some methods use large training data to cover different conditions. However, almost all methods use a facial landmark predictor to locate the position of the facial landmarks. These methods perform face alignment by fitting the geometric transformation between the positions of the predicted facial landmarks and the pre-defined landmarks. However, STN[18], proposed by DeepMind, allows the spatial manipulation of data within the network. STN learns the invariance of translation, scale, rotation, and more generic warping from the feature map itself, without extra training supervision. Because the gradient can be calculated and backward computation is possible, the whole framework becomes end-to-end trainable. The experiments show that STN can learn to transform invariance and reduce unnecessary calculations.

For the input feature map, $U \in \mathbb{R}^{H \times W \times C}$, the points can be regarded as the transformed result of the aligned feature with θ. Meanwhile, θ can have different formats for different transformation types. In the affine transformation used in our method, θ is a 2 × 3 matrix given by

$$\theta = \begin{bmatrix} \cos\alpha & -\sin\alpha & t_1 \\ \sin\alpha & \cos\alpha & t_2 \end{bmatrix} \tag{1}$$

where α indicates the rotation angle, and t_1, t_2 indicate translation. The pixels in the input feature map are the original points, which are denoted as (x_i^s, y_i^s), and the pixels in the output feature map are the target points, denoted as (x_i^t, y_i^t). Eq. (2) shows the pointwise 2D affine transformation.

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \tau_\theta \cdot l_i^t = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{2}$$

A sampling kernel is applied on each channel of the input feature map U to obtain the corresponding pixel value in the output feature map V. Using bilinear sampling kernel, the process is given as

$$V_i = \sum_{h=1}^{H} \sum_{w=1}^{W} U_{hw} \max\left(0, 1 - \left|x_i^s - w\right|\right) \max\left(0, 1 - \left|y_i^s - h\right|\right) \tag{3}$$

According to Chen et al.[45], it is easy to calculate the partial derivatives for U, x and y to reduce the loss and update the parameters. The gradients are defined as

$$\frac{\partial V_i}{\partial U_{hw}} = \sum_{h=1}^{H} \sum_{w=1}^{W} \max\ (0, 1 - |x_i^s - w|) \max(0, 1 - |y_i^s - h|) \tag{4}$$

$$\frac{\partial V_i}{\partial x_i^s} = \sum_{h=1}^{H} \sum_{w=1}^{W} U_{hw} \max(0, 1 - |y_i^s - h|) g(w, x_i^s) \tag{5}$$

$$g(w, x_i^s) = \begin{cases} 1, & x_i^s \leqslant w < x_i^s + 1 \\ -1, & x_i^s - 1 < w < x_i^s \\ 0, & otherwise \end{cases} \tag{6}$$

Similar to Eq.(5) for $\frac{\partial V_i}{\partial y_i^s}$

3.4 Recognition part

After the detection boxes areobtained, the filtered boxes are fed into the recognition part. Meanwhile, another STN is added before the recognition part to align the detected faces. ROI Pooling operator extracts the features in the detection boxes from the shared feature maps. The STN predicts the transformation parameters and applies the transformation on the region features. The whole network is end-to-end trainable and is supervised only by face bounding boxes and personal identities from publicly available datasets.

The architecture of the proposed method is shown in Figure 1. The VGG-16 network feature extractor includes 13 convolutions and outputs 512 feature maps named *conv*5. The RPN network is a fully convolutional network that outputs the candidate boxes for proposal regions. The ROI Pooling operator then extracts the features of the proposal regions from *conv*5 feature maps and resizes them to 7×7, which are the same size as the convolution output in the pre-trained VGG model. The STN for detection includes a convolution layer with $output = 20, kernel\ size = 5, stride = 1$, a pooling layer with $kernel\ size = 2$, and a fully-connected layer initialized with $weight = 0, bias = [\,1\ 0\ 0\ 0\ 1\ 0\,]^T$. The following four fully-connected layers are set to classify the proposal region and regress the bounding box. The Softmax loss layer and the L2 loss layer are used for supervising the detection training. The STN for recognition includes a convolution layer with $output = 20, kernel\ size = 5, stride = 1$, a pooling layer with $kernel\ size = 2$, and a fully-connected layer which is initialized as same.

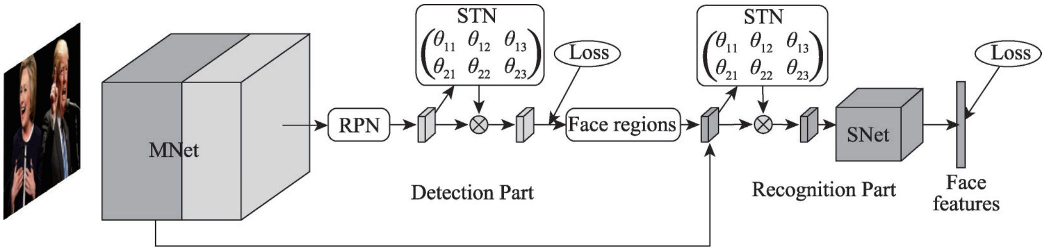
MNet
RPN
STN
$\begin{pmatrix}\theta_{11} & \theta_{12} & \theta_{13}\\ \theta_{21} & \theta_{22} & \theta_{23}\end{pmatrix}$
Loss
Face regions
STN
$\begin{pmatrix}\theta_{11} & \theta_{12} & \theta_{13}\\ \theta_{21} & \theta_{22} & \theta_{23}\end{pmatrix}$
SNet
Loss
Detection Part
Recognition Part
Face features

Figure 1 Architecture of the proposed network consists of two parts. MNet is the main network for extracting the shared features and the detection features, and SNet is a minor network for extracting facial features from shared features used to recognize a person's identity. RPN uses MNet features to generate candidate regions. STN predicts transformation parameters from the input feature. SNet uses the transformed features from shared features in detected face regions to output face features. In the proposed framework, MNet and SNetcan be any CNN network, e.g., VGG-16 architecture[42] for MNet and ResNet[43] for SNet.

As ResNetcan handle most machine learning problems, we used ResNet to extract the distinguishable facial features, due to the feature sharing, the ResNet could be simplified. It produces a 512-dimensional output feature vector that can capture stable individual features. Inspired by Wen et al.[14], we used center loss function and Softmax loss for co-operative learning of discriminative features for recognition.

4 Experiments

We performed several experiments to demonstrate the effectiveness of the proposed method. For feature sharing, we shared different convolution features to compare the decreasing loss, face recognition accuracy, and testing time. Finally, we compared our method with other methods in terms of face detection recall and face recognition accuracy.

4.1 Implementation details

The whole pipeline was implemented in Python. CNN was trained on TitanXPascal GPU using Caffe framework[4]. We used $base\_lr = 0.1, momentum = 0.9, lr\_policy = step$ with $gamma = 0.1, stepsize = 20000$. To train more faces in one batch, we set $iter\_size = 32$ and max_iter is 100000. The batch size for RPN training was 300 and the positive overlap was set to 0.7. Due to the different training targets, the two datasets produced different loss_weight for backward processing, as summarized in Table 1. The detection part was trained to produce accurate detection results in the first 50000 iterations. Meanwhile, the detection and recognition parts were both trained in the next 30000 iterations, while the detection and recognition parts were both trained in the last 20000 iterations. Furthermore, we fixed the detection network parameters and trainedonly the recognition network. The training process is shown in Figure 2.

Table 1 Loss weight of WIDER FACE and CASIA-WebFace datasets for different targets

Loss weight	RPN	Detection	Recognition
WIDER FACE (Yang et al.[19])	1.0	1.0	0.0
CASIA-WebFace (Yi et al.[50])	0.0	0.5	1.0

It took around 7 days to training the network on WIDER FACE dataset and CASIA WebFace dataset. It took around 2 days to train the first 50000 iterations fordetection training, and 3 days for co-training, 2 days to train the final recognition task. In average the detection took 150ms for image size of 1000×750. It took about 1ms for STN forwards. The feature extraction time per face is shown in Table 2.

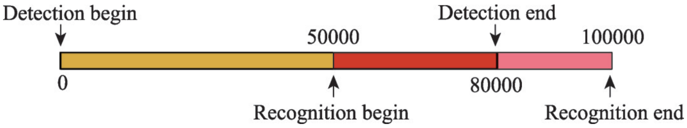

Figure 2 Training stage. Detection network is trained before 80000 iterations and recognition network is trained from 50000 iterations.

Table 2 Comparison of single face recognition model on LFW datasets

Method	LFW accuracy(%)
DeepID (Sun et al.[9])	97.45
VGGFace (Parkhiet al.[51])	97.27
Our method	**98.63**

4.2 Dataset

4.2.1 Face detection

WIDER FACE[19] training set was used for training and FDDB[21] dataset was used for testing. WIDER FACE dataset is a publicly available face detection benchmark dataset comprising 393703 labeled faces in 32203 images. FDDB contains 5171 annotated faces in a set of 2845 images taken from the Faces in the Wild dataset.

4.2.2 Face recognition

The model was trained on CASIA-WebFace dataset containing 10575 subjects and 494414 images obtained from the internet, and on LFW[22] dataset containing more than 13000 facial images obtained from the internet. Scaling up the image count in the batch can boost the generalization ability of the model. Although Faster R-CNN network framework can use any image size as the input, it loses the ability to process several images simultaneously. Some methods use the cropped images for the training efficiency. However, to retain the ability of arbitrary input, images that were randomly selected from the CASIA-WebFace dataset were combined. By stitching 12 images in 3 rows, the original image size was 250×250. For four images in each row, the final target training sample size was 1000×750. Examples of the stitched images are shown in Figure 3.

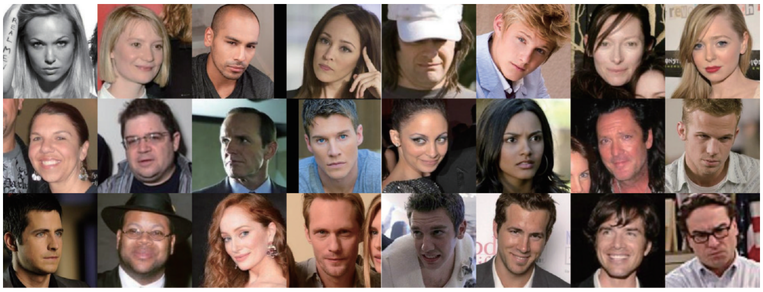

Figure 3 Additional training dataset in the proposed method.

4.3 Shared features

In common face recognition pipelines, face alignment is applied to the original input face patches. Figure 4 shows the spatial transformation results of faces, which prove that STN has the ability to learn a suitable transformation. Because STN can be insertedinto any convolution layer to align the features, we shared the features for both face detection and face recognition tasks. In the original detection framework, the first

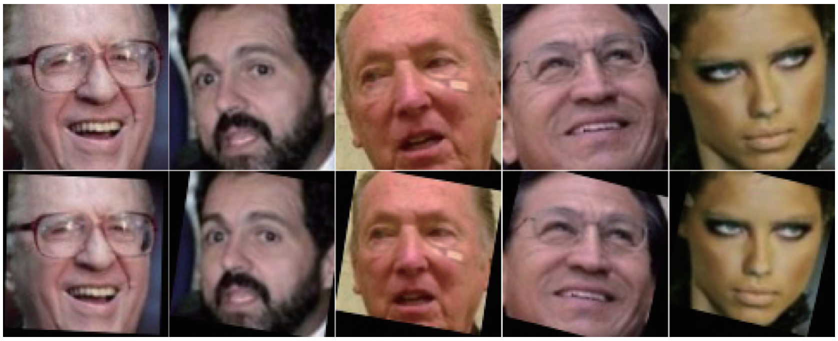

Figure 4 Spatial transformation results of CASIA-WebFace dataset. The upper figures are the original facial patches. The lower images are the transformed results.

five convolution blocks were used to extract features for the face detection task. We designed several training experiments for sharing different features and tested the results on the FDDB and LFW datasets.

Figure 5 illustrates feature sharing. The shared features include conv1, conv2, conv3, and conv4 of the VGG-16 structure. For each experiment, the recognition networks were simplified by cutting the corresponding convolution layers. The deeper the sharing layers, the smaller the recognition network became. Figure 6 shows the decreasing loss during the training stage for the shared features. It demonstrates that the deep sharing features aids rapid loss convergence. Figure 7 shows the FDDB results of different face detection algorithms, some detection results are shown in Figure 8. Table 3 shows the LFW results of different models. The model had 98.63% accuracy on LFW dataset sharing the features of conv3, which is better than using the original image patches. The accuracy decreased when sharing became deeper due to the reduced convolutional layer in the recognition for extracting the distinguishable facial features.

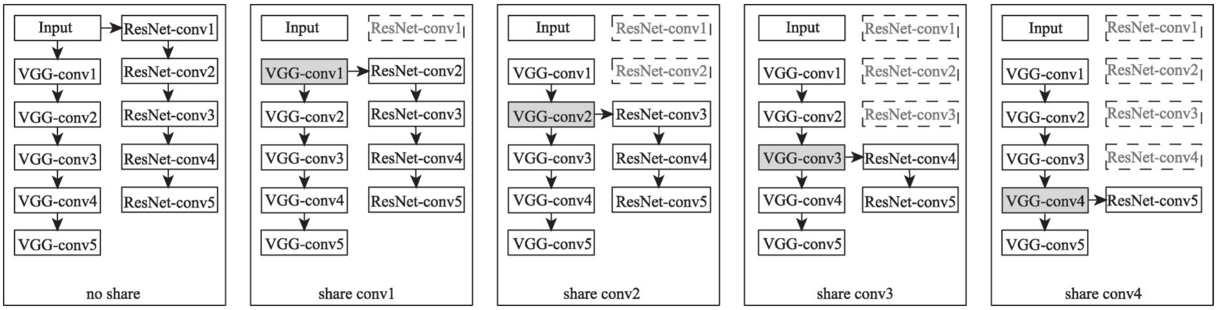

Figure 5 Shared features are illustrated in bold boxes and deleted convoluted layers in the recognition network are illustrated in dotted boxes. Based on our experiments, share conv3 outperforms other options.

4.4 Applications

Face detection and recognition are widely used in AR-based devices, such as AR glasses (Figure 9). The device captures people, displays tracking boxes near people's faces, and displays their info. It processes camera frames and feedbacks the results in real-time.

Since AR devices are portable and source-limited, model size and total multiply-add counts (MAC) should be taken into account. This method can reduce flops due to shared convolution feature maps. Furthermore, this method is flexible and combines with novel lightweight

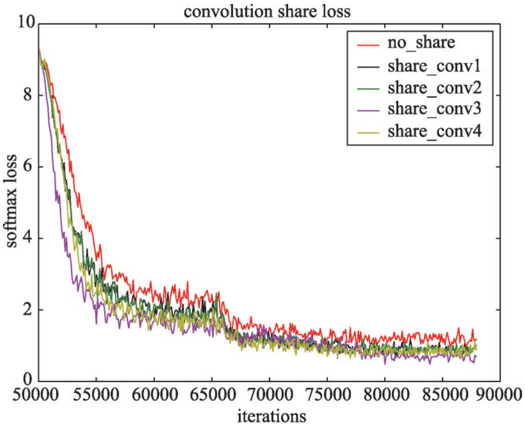

Figure 6 Recognition loss during training for different parameters.

networks[52−59] designed for source-limited platforms.

5 Conclusions

In this work, a novel end-to-end trainable framework for face detection and recognition tasks was proposed, in which STN is used to align the feature map without an additional face alignment stage. The face detection and recognition networks share a common low-level feature so that they can benefit from each other. The experiment demonstrated that STN could replace the facial landmark prediction stage, and sharing common features could result in quicker loss convergence and greater accuracy. The single model was tested on FDDB and LFW datasets. The results showed that the single model is better than two separate models.

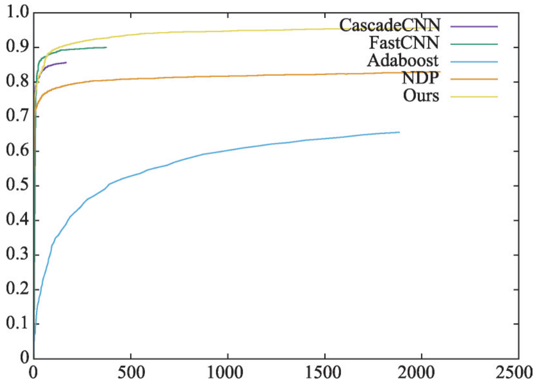

Figure 7 FDDB results compared with CascadeCNN[47], FastCNN[48], Adaboost[1], and NPD[49].

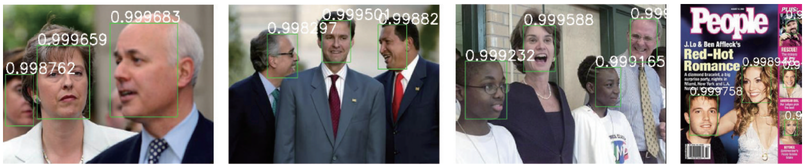

Figure 8 Examples of detection results on the FDDB dataset.

Table 3 LFW results of different share convs

	LFW results (%)	test time (ms)
No share	98.06	8.3
Share conv1	98.10	7.8
Share conv2	98.23	7.2
Share conv3	**98.63**	6.6
Share conv4	98.13	6.3

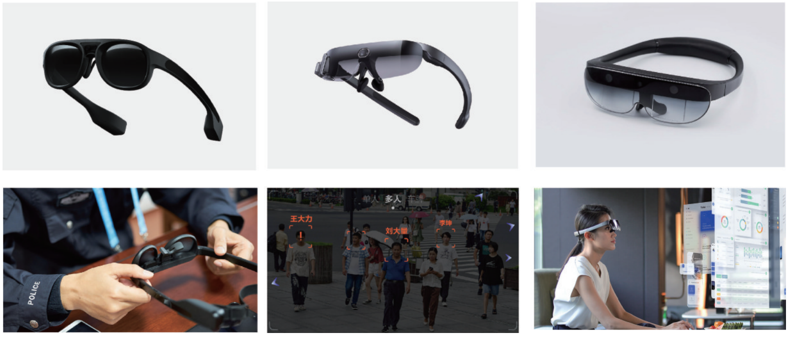

Figure 9 AR glasses and AR applications.

In the future, we will extend this method for other facial target prediction tasks and make the network size smaller to speed up both the training and testing stages.

References

1 Viola P, Jones M. Robust real-time face detection. International Journal of Computer Vision, 2004, 57(2): 137−154 DOI: 10.1023/B:VISI.0000013087.49260.fb

2 Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA, IEEE, 2016, 779‒788
DOI:10.1109/cvpr.2016.91

3 Redmon J, Farhadi A. Yolo9000: Better, faster, stronger. arXiv, 2016
DOI:10.1109/CVPR.2017.690

4 Ren S, Cao X, Wei Y, Sun J. Face alignment via regressing local binary features. IEEE Transactions on Image Processing, 25(3):1233‒1245
DOI:10.1109/TIP.2016.2518867

5 Ren S Q, Cao X D, Wei Y C, Sun J. Face alignment at 3000 FPS via regressing local binary features. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH, USA, IEEE, 2014, 1685‒1692
DOI:10.1109/cvpr.2014.218

6 Zhang K P, Zhang Z P, Li Z F, Qiao Y. Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 2016, 23(10): 1499‒1503
DOI:10.1109/lsp.2016.2603342

7 Zhang Z, Luo P, Loy C, Tang X. Facial landmark detection by deep multi-task learning. Computer Vision ECCV 2014, 2014
DOI:10.1007/978-3-319-10599-4_7

8 Deng Z P, Li K, Zhao Q J, Chen H. Face landmark localization using a single deep network//Biometric Recognition. Cham: Springer International Publishing, 2016, 68‒76
DOI:10.1007/978-3-319-46654-5_8

9 Sun Y, Wang X G, Tang X O. Deep convolutional network cascade for facial point detection. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition. Portland, OR, USA. IEEE, 2013, 3476‒3483
DOI:10.1109/cvpr.2013.446

10 Liong V E, Lu J, Wang G. Face recognition using deep PCA. In: 2013 9th International Conference on Information, Communications & Signal Processing. Tainan, Taiwan, China, 2013
DOI:10.1109/ICICS.2013.6782777

11 Lu J W, Liong V E, Wang G, Moulin P. Joint feature learning for face recognition. IEEE Transactions on Information Forensics and Security, 2015, 10(7): 1371‒1383
DOI:10.1109/tifs.2015.2408431

12 Stuhlsatz A, Lippel J, Zielke T. Feature extraction with deep neural networks by a generalized discriminant analysis. IEEE Transactions on Neural Networks and Learning Systems, 2012, 23(4): 596‒608
DOI:10.1109/tnnls.2012.2183645

13 Zhang X, Fang Z Y, Wen Y D, Li Z F, Qiao Y. Range loss for deep face recognition with longtail. arXiv, 2016
DOI:10.1109/ICCV.2017.578

14 Wen Y D, Zhang K P, Li Z F, Qiao Y. A Discriminative Feature Learning Approach for Deep Face Recognition. Cham: Springer International Publishing, 2016, 499‒515
DOI:10.1007/978-3-319-46478-7_31

15 Ranjan R, Patel V M, Chellappa R. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. arXiv, 2016

16 Ranjan R, Sankaranarayanan S, Castillo C D, Chellappa R. An all-in-one convolutional neural network for face analysis. In: 12th IEEE International Conference on Automatic Face & Gesture Recognition. IEEE, 2017, 17‒24
DOI:10.1109/FG.2017.137

17 Ren S Q, He K M, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137‒1149
DOI:10.1109/tpami.2016.2577031

18 Jaderberg M, Simonyan K, Zisserman A, Kavukcuoglu K. Spatial transformer networks. In: Advances in neural information processing systems, 2015

19 Yang S, Luo P, Ploy C C, Tang X. Wider face: A face detection benchmark. In: Computer Vision and Pattern Recognition (CVPR). 2016, 5525‒5533

DOI:10.1109/CVPR.2016.596

20 Yi D, Lei Z, Liao S, Li S Z. Learning face representation from scratch. arXiv preprint arXiv:1411.7923, 2014

21 Jain V, Learned-Miller E G. Fddb: A benchmark for face detection in unconstrained settings. UMass Amherst Technical Report, 2010
DOI:10.1007/978-3-319-46448-0_2

22 Huang G B, Ramesh M, Berg T, Learned-Miller E. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. University of Massachusetts, Amherst, 2007

23 Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C Y, Berg A C. Ssd: Single shot multibox detector. Cham: Springer, 2016, 21–37

24 Dai J F, Li Y, He K M, Sun J. R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems. 2016, 379–387

25 Girshick R, Donahue J, Darrell T, Malik J. Region-based convolutional networks for accurate object detection and segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(1): 142–158
DOI:10.1109/tpami.2015.2437384

26 Girshick R. Fast r-CNN. In: Proceedings of the IEEE international conference on computer vision. IEEE, 2015, 1440–1448
DOI:10.1109/ICCV.2015.169

27 Zheng Y, Zhu C, Luu K, Bhagavatula C, Le T H N, Savvides M. Towards a deep learning framework for unconstrained face detection. In: IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS). IEEE, 2016
DOI:10.1109/BTAS.2016.7791203

28 Jiang H, Learned-Miller E. Face detection with the faster R-CNN. In: 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017). IEEE, 2017, 650–657
DOI:10.1109/FG.2017.82

29 Sun X D, Wu P C, Hoi S C H. Face detection using deep learning: an improved faster RCNN approach. Neurocomputing, 2018, 299: 42–50
DOI:10.1016/j.neucom.2018.03.030

30 Hashimoto K, Xiong C, Tsuruoka Y, Socher R. A joint many-task model: Growing a neural network for multiple NLP tasks. arXiv preprint arXiv:1611.01587, 2016
DOI:10.18653/v1/D17-1206

31 He R, Lee W S, Ng H T, Dahlmeier D. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. arXiv preprint arXiv:1906.06906, 2019
DOI:10.18653/v1/P19-1048

32 Lu Y X, Kumar A, Zhai S F, Cheng Y, Javidi T, Feris R. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, IEEE, 2017, 5334–5343
DOI:10.1109/cvpr.2017.126

33 Khamis S, Kuo C H, Singh V K, Shet V D, Davis L S. Joint learning for attribute-consistent person re-identification. In: Agapito L, Bronstein M M, Rother C, editors, Computer Vision-ECCV 2014 Workshops. Cham: Springer International Publishing, 2015, 134–146
DOI:10.1007/978-3-319-16199-0_10

34 Chen D, Ren S, Wei Y, Cao X, Sun J. Joint cascade face detection and alignment. In: European Conference on Computer Vision. Springer, 2014, 109–122
DOI:10.1007/978-3-319-10599-4_8

35 Deng J, Guo J, Zhou Y, Yu J, Kotsia I, Zafeiriou S. Retinaface: Single-stage dense face localisation in the wild. arXiv preprint arXiv:1905.00641, 2019

36 Bartz C, Yang H, Meinel C. STNOCR: A single neural network for text detection and text recognition. arXiv preprint arXiv:1707.08831, 2017

37 Cheng Z, Liu X, Bai F, Niu Y, Pu S, Zhou S. Arbitrarily-oriented text recognition. CoRR, abs/1711.04226, 2017

DOI:10.1109/CVPR.2018.00584

38 Luo H, Fan X, Zhang C, Jiang W. Stnreid: Deep convolutional networks with pairwise spatial transformer networks for partial person re-identification. IEEE Transactions on Multimedia, 2020
DOI:10.1109/TMM.2020.2965491

39 Fang H, Xie S, Lu C. RMPE: regional multi-person pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision. IEEE, 2017, 2334‒2343
DOI:10.1109/ICCV.2017.256

40 Zhong Y, Chen J, Huang B. Towards end-to-end face recognition through alignment learning. IEEE signal processing letters, 2017, 24(8), 1213‒1217
DOI:10.1109/LSP.2017.2715076

41 Wu W L, Kan M N, Liu X, Yang Y, Shan S G, Chen X L. Recursive spatial transformer (ReST) for alignment-free face recognition. In: 2017 IEEE International Conference on Computer Vision. Venice, IEEE, 2017, 3772‒3780
DOI:10.1109/iccv.2017.407

42 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv: 1409, 2014

43 He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Computer Vision and Pattern Recognition (CVPR). IEEE, 2016, 770‒778

44 Deng J, Dong W, Socher R, Li L J, Li K, Li F. Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition (CVPR), 2009, 248‒255
DOI:10.1109/CVPR.2009.5206848

45 Chen D, Hua G, Wen F, Sun J. Supervised transformer network for efficient face detection. In: European Conference on Computer Vision. Cham: Springer, 2016, 122‒138
DOI:10.1007/978-3-319-46454-1_8

46 Jia Y Q, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T. Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia. ACM, 675‒678

47 Li H X, Lin Z, Shen X H, Brandt J, Hua G. A convolutional neural network cascade for face detection. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA, USA, IEEE, 2015, 5325‒5334
DOI:10.1109/cvpr.2015.7299170

48 Triantafyllidou D, Tefas A. A fast deep convolutional neural network for face detection in big visual data. Advances in Big Data, 2017, 61‒70
DOI:10.1007/978-3-319-47898-2_7

49 Liao S, Jain A K, Li S Z. A fast and accurate unconstrained face detector. IEEE transactions on pattern analysis and machine intelligence, 2015, 38(2), 211‒223

50 Sun Y, Wang X, Tang X. Deep learning face representation from predicting 10000 classes. In: Computer Vision and Pattern Recognition (CVPR). IEEE, 2014, 1891‒1898
DOI:10.1109/TPAMI.2015.2448075

51 Parkhi O M, Vedaldi A, Zisserman A. Deep face recognition. In: BMVC, 2015
DOI:10.5244/C.29.41

52 Tan M, Le Q V. Efficientnet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv: 1905.11946, 2019

53 Sandler M, Howard A G, Zhu M, Zhmoginov A, Chen L. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. arXiv preprint arXiv:1801.04381, 2018
DOI:10.1109/CVPR.2018.00474

54 Howard A G, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017

55 Howard A, Sandler M, Chu G, Chen L, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V, Le Q V, Adam H. Searching for mobilenetv3. CoRR, abs/1905.02244, 2019. 8
DOI:10.1109/ICCV.2019.00140

56 Wu B, Dai X, Zhang P, Wang Y, Sun F, Wu Y, Tian Y, Vajda P, Jia Y, Keutzer K. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2018, 10726−10734
DOI:10.1109/CVPR.2019.01099

57 Chen S, Liu Y P, Gao X, Han Z. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In: Chinese Conference on Biometric Recognition. Cham: Springer, 428−438
DOI:10.1007/978-3-319-97909-0_46

58 Tan M, Chen B, Pang R, Vasudevan V, Le Q V. Mnasnet: Platform-aware neural architecture search for mobile. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2018, 2815−2823
DOI:10.1109/CVPR.2019.00293

59 Zhang X, Zhou X, Lin M, Sun J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2017, 6848−6856
DOI:10.1109/CVPR.2018.00716