IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 15, 2019, accepted January 3, 2020, date of publication January 8, 2020, date of current version January 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964838

A CPU Real-Time Face Alignment for Mobile Platform

XIN NING[1,2,3,4], (Member, IEEE), PENGFEI DUAN[4], WEIJUN LI[1,2,3], (Senior Member, IEEE), YUAN SHI[4], AND SHUANG LI[1,2,3], (Member, IEEE)

[1]Institute of Semiconductors, Chinese Academy of Sciences, Beijing 100083, China
[2]Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China
[3]School of Microelectronics, University of Chinese Academy of Sciences, Beijing 100049, China
[4]Cognitive Computing Technology Joint Laboratory, Wave Group, Beijing 102208, China

Corresponding authors: Xin Ning (ningxin@semi.ac.cn) and Weijun Li (wjli@semi.ac.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572458 and Grant 61901436.

ABSTRACT Face alignment is a common technology in face recognition and face verification field. Previous works mostly pay attention to improving the accuracy of prediction and ignored the practicability of the method. In this paper, we aim at providing a two-stage face alignment network for mobile platform. Firstly, the network was trained with residual label which is the difference between ground truth and mean shape. Secondly, the input data in the second stage is composed of the original data and generated heatmap which enriched the data types. Finally, a new loss function is used to enhance the convergence of local region. Experimental results show that our method not only provides high precision but also improve the real-time processing performance on the mobile platforms.

INDEX TERMS Residual label, heatmap, global pooling, loss function.

I. INTRODUCTION

Face alignment refers to the location of the key parts of a given face image, which is one of the key steps in face recognition [46], [47], [48], face verification [18], [19], [25], face beautification and other applications.

Before the rapid development of deep neural network technology, most face alignment algorithms use classical machine learning algorithms to locate key points such in [3], [4], [6], [8], [13]. The advantages of these methods are fast operation speed and small model. For example, Cootes *et al.* [38], [39] presented active appearance models (AAM) to control modes of shape which construct an efficient iterative matching algorithm by learning the relationship between perturbations in the model parameters and the induced image errors. Cristinacce *et al.* [50] presented an efficient and robust model matching method which uses a joint shape and texture appearance model to generate a set of region template detectors and improves localization accuracy on two datasets. Lee *et al.* [7] proposed a method that combined gaussian process regression trees (GPRT) in a cascade stage to get accurate key locations. Kowalski and Naruniec [40] presented a face alignment pipeline based on novel k-cluster regression forests with weighted splitting and showed state-of-the-art results at that time. Tuzel *et al.* [41] proposed a cascade in which each stage consists of a mixture of regression experts that learns a customized regression model and achieved the current optimal results.

Last few years, with the development of deep learning technology, more people pay attention to applying deep learning technology to solve face alignment tasks [9, 17, 21, 25]. For instance, Bhagavatula *et al.* [1] proposed a method simultaneously extract the 3D shape and the semantically consistent 2D alignment, which achieved current optimal results of 3D face alignment. Zhou *et al.* [42] presented a cascade coarse-to-fine convolutional network to localize extensive facial landmarks and got high accuracy. Kumar and Chellappa [49] designed a pose conditioned dendritic convolution neural network (PCD-CNN), where a classification network is followed by a second and modular classification network, which trained in an end-to-end fashion to obtain accurate landmark points. Kowalski *et al.* [10] presented a deep alignment network (DAN) which contains multiple types of feature and achieved state-of-the-art at that time. Bulat and Tzimiropoulos [45]

The associate editor coordinating the review of this manuscript and approving it for publication was Peter Peer.

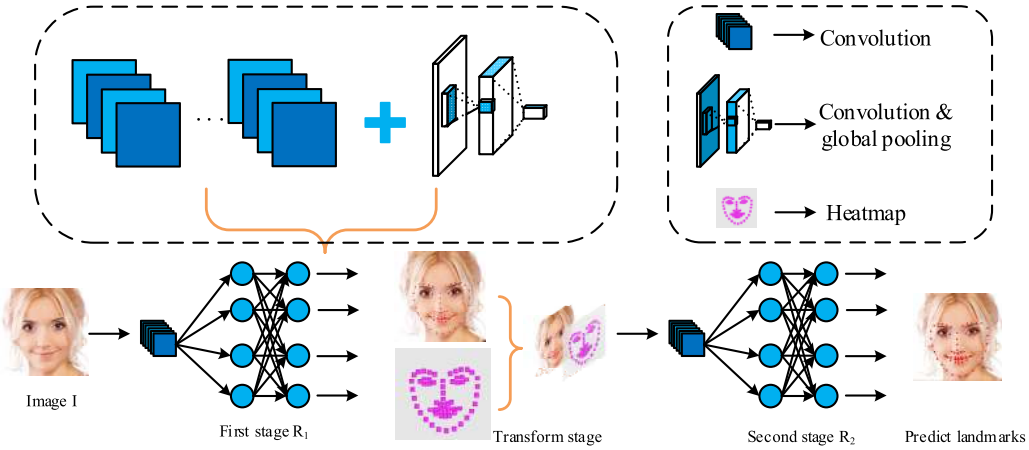

FIGURE 1. The illustration of our architecture (The architecture mainly consists of two network stage and a transform stage).

proposed a method which built upon the idea of convolutional part heatmap regression and ranked 1st in the 3D Face Alignment in the Wild (3DFAW) Challenge. Wu *et al.* [9] proposed that face boundary information should be used to enhance the learning of network features and a discrimination network is used to judge the quality of the heatmap. Finally, the key points are obtained by regression network.

In addition, there are some approaches to improve the speed of the algorithms. Ren *et al.* [6] proposed a locality principle for feature extraction called learning binary features (LBF), which realized 300 FPS on a mobile phone. Guo *et al.* [34] proposed a practical facial landmark detector (PFLD), which was cutting-edge and exhibited remarkable performance, but it was still unable to achieve real-time capabilities on the low-power mobile platform.

From the introduction of face alignment technology to the present, most researchers pay attention to improving the positioning accuracy of the algorithm and have neglected the problems in the application of the algorithm, such as being time-consuming and having a large model.

To this end, we proposed a CPU real-time face alignment approach for mobile terminals and 90 FPS can be achieved on the ARM platform. Therefore, it can carry out real-time and accurate face key points location in the actual product.

Based on our previous work [52], in order to solve the problem of some local region is difficult to converge, a new loss function is used to enhance the convergence of local region.

To summarize, the three main contributions of this work are as follows:

- The residual between ground truth and mean shape is used as the training label, and the joint heatmap and original map as the input of second-stage deepens the feature learning of the landmarks.
- A two-stage face alignment network is proposed using the convolution and global pooling structure with real-time ARM speed of 90 FPS.
- A new loss is used to train the proposed method that can solve the problem of inaccurate location of some parts.

The paper will be presented in four chapters. First, we introduce the research status of face alignment and structure of the paper. In the second part, we describe the algorithm flow and structure of the proposed method. The third part is the comparison experiments between our method and the previous methods in the operation platform and prediction accuracy. The last part is the summary and conclusion.

II. PROPOSED METHOD

In this section, the illustration of our architecture is showed in Fig. 1. Two network stages are consisted of convolution, global pooling, activation layers and BN layers. The role of the transform stage is to calculate predict landmarks of first stage, training label and heatmap of the second stage.

Firstly, we discuss the data fusion with heatmap and its advantages with training network. Then, we focus on the proposed network and training labels in the method. Finally, we present the details of how the new loss function work together with the network to refine the predict landmarks.

In addition, the procedure of the proposed method is summarized in algorithm1.

A. DATA FUSION WITH HEATMAP

The heatmap has corresponding applications in the field of computer vision such as population density estimation, human posture judgment and pedestrian detection. Therefore, we also try to apply it to face alignment to strengthen the characteristics of key parts. We designed to generate a heatmap in the second stage and combine it with the original input. A landmark heatmap is an image with high-intensity values around landmark locations where intensity decreases with distance from the nearest landmark. The generation of the heatmap depends on the prediction value generated in the first stage, and the generation of the prime value of the heatmap is

Algorithm 1 Steps for Training and Testing

Require: Input image I, landmark coordinates S_{gt}, mean shape S_m, first regression network R_1, second regression network R_2.

1: **while** the accuracy of landmarks predicted by R_1 in validation set stops in the first stage **do**;
2: Forward R_1 to get output of first stage S_{out1} and calculate loss of first stage by $|_{gt}$- S_m-$S_{out1}||_2$;
3: Using Gradient descent method to optimize R_1 by minimizing $|_{gt}$- S_m-$S_{out1}||_2$; **end while**
4: Calculate predict landmarks of first stage, training label of second stage and heatmap H in the transform stage;
5: Concat I and H as the input of second stage;
6: **while** the accuracy of landmarks predicted by $R_1 + R_2$ in validation set stops in the second stage **do**;
7: Forward $R_1 + R_2$ to get output of second stage S_{out2} and calculate loss of second stage by $|_{gt}$- S_m-S_{out1}-$S_{out2}||_2$;
8: Using Gradient descent method to optimize R_1+R_2 by minimizing $|_{gt}$- S_m-S_{out1}-$S_{out2}||_2$; **end while**
9: Output predict landmarks equal to ($S_{out1} + S_m + S_{out2}$);

FIGURE 2. Data and heatmap.

derived from Equation (1) in the transform layer.

$$H(x, y) = \frac{1}{1 + \min_{s_i \in s_{pre1}} \|(x,y) - s_i\|}, \tag{1}$$

where $H(x, y)$ denotes the pixel value of the generated heatmap, while S_i denotes the coordinates of the predicted landmarks obtained in the first stage. While $\|.\|$ designates the absolute deviation between the pixel and the i-th landmark. S_{pre1} is a vector of predicted result after stage 1. The value of S_{pre1} is equal to the sum of S_{out1} and S_m. The heatmap values are calculated in a circle of radius 8 around each landmark. The closer the coordinates of the heatmap value are to the landmarks, the closer the pixel value is to 1. On the contrary, the farther the coordinates are from the landmarks, the closer the pixel value is to 0. The generated heatmap is shown in Fig 2. The reason as to why the radius is 8 is because this is a parameter for generating a heatmap of key points of the human face. If the radius is too large, the information of heatmap is too redundant, and if the radius is too small, the information of heatmap is too little, which is of little use to the diversity of input information in the network.

TABLE 1. Parameters of the proposed network. The format according to height * width * depth, stride.

Name	Shape-in	Shape-out	Kernel
Conv1_1	112*112*1	56*56*16	3*3*1,2
Conv1_2	56*56*16	28*28*32	3*3*16,2
Pool1_2	28*28*32	14*14*32	2*2*32,2
Conv1_2.1	14*14*32	7*7*64	3*3*32,2
Pool1_2.1	7*7*64	1*1*64	global_pool
Conv1_3	14*14*32	7*7*64	3*3*32,2
Pool1_3	7*7*64	4*4*64	2*2*64,2
Conv1_3.1	4*4*64	2*2*64	3*3*64,2
Pool1_3.1	2*2*64	1*1*64	global_pool
Conv1_4	4*4*64	2*2*64	3*3*64,2
Pool1_4.1	2*2*64	1*1*64	global_pool
Concat	1*1*64*3	1*1*192	----
Fc_1	1*1*192	1*136	----
Transform	1*136	112*112*1	----
Concat	112*112*1*2	112*112*2	----
Conv2_1	112*112*2	56*56*8	3*3*2,2
Pool2_1	56*56*8	28*28*8	3*3*8,2
Conv2_2	28*28*8	28*28*16	3*3*8,1
Pool2_2	28*28*16	14*14*16	3*3*16,2
Conv2_2.1	14*14*16	14*14*64	3*3*16,1
Pool2_2.1	14*14*64	1*1*64	global_pool
Conv2_3	14*14*16	7*7*64	3*3*16,2
Pool2_3	7*7*64	3*3*64	3*3*64,2
Conv2_3.1	3*3*64	3*3*64	3*3*64,1
Pool2_3.1	3*3*64	1*1*64	global_pool
Conv2_4	3*3*64	2*2*64	3*3*64,2
Pool2_4.1	2*2*64	1*1*64	global_pool
Concat	1*1*64*3	1*1*192	----
Fc_2	1*1*192	1*136	----

B. PROPOSED NETWORK

When designing the network, according to some experience, the larger the feature map, the related convolution and pooling will be more time-consuming. Therefore, we did not set a large input and intermediate feature map when setting the parameters of the network structure. The convolution layer generally uses a 3∗3 convolution kernel. The specific parameter settings are shown in Table 1.

In the proposed network, the input of the network is set to 112 ∗ 112, because too large image input will cause more time to be consumed in the first layer of convolution at every stage, on the contrary, too small input will cause less information. The combination of convolution and pooling will take less time than multi convolution layers, and it can save the fine-grained features. Several pooling layers need to be connected at the end of each stage. After the connection operation, there will be a full connected layer. Because the output size of the pooling layer will affect the size of the whole model, we suggest setting the output size of the pooling layer to 64, so that the model size will have certain advantages in the migration and loading on the mobile platform. The heatmap is generated by the output results of key points in the first stage and transferred to the second stage as input of

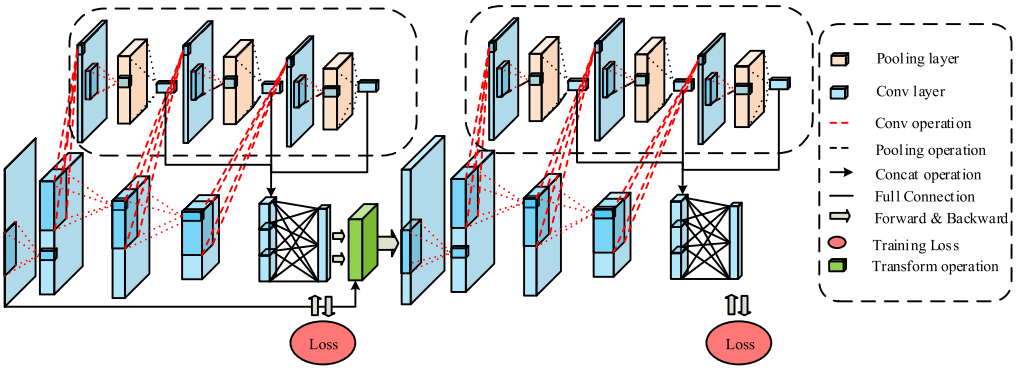

FIGURE 3. Proposed network (The heatmap is generated in the transform layer, the second stage begins training after the first stage of training is completed.).

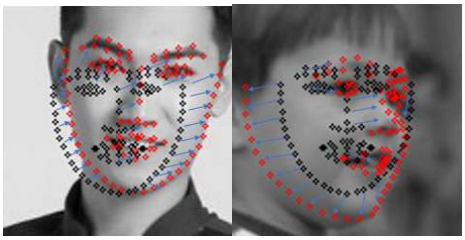

FIGURE 4. Image of residual regression.

the second stage. As a way to increase the representation of features, the heatmap increases the kinds of features learned by the network of second stage and strengthens the local feature information of the key points. The convolution and pooling structure allow the network to be faster and exhibit good learning ability. Therefore, the proposed two-stage network structure can give consideration to both time and good expression ability.

C. TRAINING LABEL WITH RESIDUAL

The development of face alignment starts from the earliest Active Appearance Models [38], [39] and Constrained Local Models [50], [51], moving to Cascaded Shape Regression (CSR) [4], [3], [6], [7], [40], [41] and deep learning methods [42], [43], [20], [44], [45]. The traditional face alignment cascade regression methods have some research results which show that the good face shape initialization can improve the regression results [4], [8], [20], [36], [37], so we use the difference between the ground truth and mean shape as the learning label of the first stage. Simultaneously, this approach has a similar application in deep learning method recent years, such as DAN [10]. In DAN, the input image is transformed for each stage so that the current estimates of the landmarks are aligned with the canonical face shape. This normalization step allows the further stages of DAN to be invariant to a given family of transforms [10]. However, our method is different from DAN, we directly adopt the difference between the ground truth and the predicted value (which equal to the sum of the output value in the first stage and the mean shape) as the training label in the second stage.

As for mean shape which is an average shape of face image placed in the detection box returned by the face detector [3], [6], [15].

The formulas for calculating the training labels of the first and second stages are as follows. The second stage begins training after the first stage of training is completed. The training label used in the second stage is the difference between the ground truth and the predicted value after the first stage, as shown in Equation (3).

$$res1 = S_{gt} - S_m, \tag{2}$$

$$res2 = S_{gt} - (S_{out1} + S_m), \tag{3}$$

where S_{gt} is a vector of ground truth landmark locations, S_m is a vector of standard landmark locations, and S_{out1} is the output result of stage 1. The value of S_{pre1} is equal to the sum of S_{out1} and S_m.

D. LOSS FUNCTION

The design of loss function plays an important role in the quality of network training, especially when the training datasets contain different categories of features and data imbalances. L1 norm loss function, also known as least absolute deviation, L2 norm loss function, also known as least square error. L1 and L2 losses are commonly used to calculate the loss of predicted values and ground truth values, and the penalties for each key point are then similar, often leading to some points being difficult to learn and not able to easily converge. Therefore, this loss function focuses on the key points that converge poorly in the process of network learning. By increasing its penalty coefficient, it has a greater gradient in the back propagation. Thus, the expression and generalization ability of the network are improved. Mathematically, the loss can be written in the following general form:

$$Loss = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{K} w_i(\left\| d_i^n \right\|_2^2), \tag{4}$$

TABLE 2. Size of each Dataset.

Datasets	Total	Train	Test	
300W	3837	3148	common	challenging
			554	135
COFW	1852	1345	507	
AFLW	24386	20000	Full	Frontal
			4386	1314

in which,

$$w_i = \begin{cases} h_w & if \ \|d_i\|_2^2 > \frac{\sum_{i=1}^{N} \|d_i\|_2^2}{N}, \\ 1 & otherwise, \end{cases} \quad (5)$$

where $\left\|d_i^n\right\|_2$ designates the distance/error of the i-th landmark of the n-th input, $\|.\|_2$demotes the Euclidean distance, K is the number of landmarks per face to detect, N denotes the number of training images in each process, w_i denotes the weight of each point, and h_w denotes the ratio of rich samples to rare samples in unbalanced sample datasets. For example, the data of opening eyes in the training dataset is about five times that of closing eyes, and the h_w is set to 5. The result of training will be better for closing eyes.

III. EXPERIMENTS

Datasets: We evaluated our method on three challenging datasets including 300W [12], COFW [26], and AFLW [31].

300W dataset [12] is a widely used, open benchmark dataset. The full test set consists of a common subset and a challenging subset. COFW dataset originally has 29 manually annotated landmarks. The COFW-68, which has been re-annotated with 68 landmarks by [27], is used in the present study. In this way, the results can be easily compared with the previous methods. AFLW dataset [31] contains face images with a large variety in appearance and environmental conditions. The size of each dataset is listed in Table 2.

Crop all the training images and resize them to 112×112 according to the provided bounding boxes, because too large image input will cause more time to be consumed in the first layer of convolution at every stage, on the contrary, too small input will cause less information. We cut the detected face by rotation, scaling and mirroring and expand the image of the original dataset into 20 images as the training set, because the images generated by this operation can contain more posture angles and face sizes. Therefore, the network has strong robustness to the location of the face detection box, even if there is a certain deviation in the position of the face detection box or the detection box is a little bit small, the landmarks of the face returned by the network are still accurate.

Evaluation metric: Following most previous works [8], [9], [20], [34], normalized root-mean-square error (NRMSE) is employed to measure the accuracy, which averages normalized errors over all annotated landmarks.

In this paper, the results are presented using two metrics. The mean distance between the predicted landmarks and the ground truth landmarks is divided by the inter-pupil distance (distance between the eye centers) or inter-ocular distance (distance between the outer eye corners) [8, 5, 6].

TABLE 3. Compared test error with different training parameters.

Batches	Learning rate	hw	300W	COFW
8	0.0001	1	6.34	6.61
8	0.0001	5	6.02	6.25
8	0.0001	10	6.05	6.57
8	0.001	5	6.19	6.83
8	0.0001	5	6.02	6.25
8	0.00001	5	7.53	7.97
8	0.0001	5	6.02	6.25
16	0.0001	5	5.91	6.19
32	**0.0001**	**5**	5.91	**6.15**
64	0.0001	5	**5.90**	6.15

The formula for calculating the normalized root-mean-square error (NRMSE) of the model is shown as follows:

$$NRMSE = \frac{\frac{1}{L}\sum_{i=1}^{L}\left\|p_i^{pre} - p_i^{gt}\right\|_2}{\left\|p_{leye} - p_{reye}\right\|_2}, \quad (6)$$

where P_i^{pre} and P_i^{gt} accordingly denote the i-th landmark coordinates of the predicted and ground-truth facial landmark positions. P_{leye} and P_{reye} denote the pupil or outer corner locations of the left eye and the right eye, respectively. L denotes the number of landmarks. Finally, the NRMSEs are averaged for all testing face samples in the present experiments as the averaged error comparisons for evaluation.

In addition, the cumulative error distribution (CED) curve is also used to compare the methods. Besides the accuracy, the processing speed is also compared.

The models and approaches are evaluated on a 3.10 GHz Intel Core i5-4440 CPU and 1.8 GHz Cortex-A17 ARM (which used in RK3288) and 2.0 GHz Cortex-A57 ARM (which used in NVIDIA Jetson TX2). NVIDIA Jetson TX2 is a popular mobile device used in many areas of artificial intelligence. We abbreviate three different CPU processors as I5, A17 and A57.

A. COMPARISON WITH DIFFERENT TRAINING HYPERPARAMETERS

In order to verify the settings of parameters in the training of network and the new loss function, the model was trained on two datasets and compared. In Table 3, the results are similar when the batch size is set to 32 and 64. When batch size is smaller, the prediction results of the model will be slightly worse. Larger ones are more suitable for the convergence of network, but too large of a batch will lead to too slow training of the network. Therefore, choosing a suitable batch is important.

For the learning rate parameter, it was determined that the convergence of the network is the best when set to 0.0001. If it is set slightly smaller, the convergence of the network will fall into local optimum, and the results are not accurate enough. In common datasets, such as 300W and COFW, it is found that large pose face images are only a small part of them,

TABLE 4. Comparing the results of different training models (300W_Fullset).

Methods	NRMSE	I5	A17	A57
One_stage	9.73	1.4	13	6
Without_conv_pool	7.17	1.7	15	8
Without_heatmap	6.79	1.7	17	9
Ours_gt	8.26	2	21	11
Shufflenet-v2	6.79	5	52	29
Mobilenet-v2	6.51	15	135	72
Ours_res [52]	**6.25**	**2**	**21**	**11**

TABLE 5. Comparing the results of different training models (COFW_68).

Methods	NRMSE	I5	A17	A57
One_stage	9.79	1.4	13	6
Without_conv_pool	7.44	1.7	15	8
Without_heatmap	7.18	1.7	17	9
Ours_gt	8.51	2	21	11
Shufflenet-v2	7.23	5	52	29
Mobilenet-v2	7.09	15	135	72
Ours_res [52]	**6.41**	**2**	**21**	**11**

making it difficult to converge well for this part of scarce data. As a consequence, the loss is increased for the key points with larger errors in the phase of network backpropagation, allowing it to converge faster in the right direction. From the results, h_w parameter set to 5 achieves the best result. If other datasets are used for training, this parameter can be adjusted accordingly.

B. COMPARISON WITH DIFFERENT TRAINING LABELS AND NETWORKS

In this paper, the data enhancement method is used to enhance the training data of the network to obtain more training data with different postures and different face sizes. The input graph of the network is the extended data of the original image data after face detection and rotation, scaling and translation operations. After expansion, the data size is 112×112, and the data preprocessing method includes subtracting the average value of 127.5 and multiplying by the scale factor size of 0.0078125. The network training batch is 32, the learning rate is set to 0.0001, and the weight attenuation is set to 0.0005. The test data set is 300W and COFW_68, and the model test within the present study implements the same face detection algorithm. Because inter-pupil normalization is very common in many papers. Therefore, in this section, this metric is used for comparation experiments. The test results are shown in Tables 4 and 5, where the number of frames processed per second is represented by FPS.

Model description:

1) **One_stage**is a model that was trained using the one stage network.
2) **Without_conv_pool**is a model that was trained using the two stages network without convolution and global pooling structure.
3) **Without_heatmap**is a model that was trained using the two stages network without heatmap in the second stage.
4) **Ours_res** is a model that was trained using the two stages network with convolution and global pooling structure.
5) **Ours_gt** is a model that was trained using the two stages network with convolution and global pooling structure but using a different training label.

The first three models mentioned above were trained using the residual error between ground truth and mean shape as label, while the last model directly takes the real ground truth landmark of the image as the training label. In addition, the models are compared with other network architectures [32], [33], such as ShuffleNet v2 and MobileNet v2. It should be noted that all units of time are milliseconds (ms) in the following tables.

From the results of the tables, it can be seen that the positioning accuracy of the two-stage network model is higher than that of the one-stage model. The reason may be that the one-stage network only preliminarily learns the amount difference between the ground truth and the mean shape, but the network learning is not precise enough, and it is difficult to get in place in one step. Therefore, the two-stage network model has certain advantages in the positioning results. When there is no convolution and global pooling structure, the time is reduced by 22%, but the accuracy is also reduced by 14:16%.

It can be seen from the comparison of the tables that the model with the residual as the label has better test performance. The reason may be that the network using the ground truth value as a label needs to be initialized from 0 and then learn the coordinates of a landmark. This is harder to learn for the network. Directly learning the residual between the ground truth and the average shape is equivalent to giving the network a standard landmark initial value. The closer the initial value of the network is to the true value, the easier the network will converge.

In order to verify the effect of heatmap on network regression learning, we compared it with the network model without heatmap. we can see from the comparison results in the tables that the network with heatmap has better performance and only increases a short time. The network with heatmap achieved 7.9:10% improvement in terms of mean error and only lost a slight amount of time.

Compared with other lightweight networks [32, 33], the network performance exhibited in the present study has improved 4-8 times, and the network test error is smaller. Analyzing the reasons, it is regarded that the present network is a two-stage cascade regression, which can gradually reduce the prediction error. This method has great advantages for key points prediction. However, the lightweight network is not a multi-stage cascade network, making it difficult to accurately locate the key points at one time.

C. COMPARISON WITH EXISTING APPROACHES

In applications such as face recognition and face verification, the face alignment algorithm is more like a pre-processing step. Since the time consumption is too long, it will directly

TABLE 6. Compared test error on 300W_Fullset.

Methods	NRMSE	I5	A17	A57
Inter-pupil Normalization				
ESR [3]	7.58	8.3	81	43
SDM [4]	7.52	14.2	126	71
LBF [6]	6.32	3.1	27	15
CFSS [8]	5.76	40	365	174
DAN_menpo [10]	5.27	150	850	416
LAB [9]	4.12	1500	-	-
PFLD [34]	3.95	11	87	42
Ours_res [52]	6.25	2	21	11
Ours_res_loss	**5.91**	**2**	**21**	**11**
Inter-ocular Normalization				
LAB [9]	3.49	1500	-	-
DAN_menpo [10]	3.44	150	850	416
SAN [35]	3.98	2500	-	-
PFLD [34]	3.40	11	87	42
Ours_res [52]	5.17	2	21	11
Ours_res_loss	**4.93**	**2**	**21**	**11**

TABLE 7. Compared test error on COFW_68.

Methods	NRMSE	I5	A17	A57
HPM [27]	6.72	3000	-	-
RCPR [26]	8.76	66	540	291
TCDCN [28]	7.66	18	161	77
CFSS [8]	6.28	40	368	174
LAB [9]	4.62	1500	-	-
Ours_res [52]	6.41	2	21	11
Ours_res_loss	**6.15**	**2**	**21**	**11**

affect the user experience of the product, so the algorithm that satisfies the speed and precision is more practical. Therefore, we will compare the running speed and accuracy with other current algorithms on different platforms.

For dataset COFW-68, we follow [9] to use inter-ocular normalization for evaluation. Because of large pose face images, we follow [8] to use face size as the normalizing factor in the AFLW [31] dataset. All the methods listed in Table 7 are evaluated using the same metric. The time in the tables is estimated from the original time of paper to i5 CPU and ARM. .

CED curves are provided to evaluate the accuracy difference in Fig. 5. The method proposed in the present study is compared against the state-of-the-art methods on the 300W dataset using CED curves. The results show that the method of PFLD and LAB have less mean normalized error, and is followed by the present approach. The prediction errors below 0.05 are nearly 70% and have good stability.

Besides the accuracy, the processing speed is also compared. As can be seen from the comparison, the present method is remarkably faster than other approaches in CPU times, and meeting the real-time requirements of mobile terminals. In addition, a test is also performed on a Cortex-A17 and Cortex-A57 ARM processor. The details of time on Intel CPU and ARM are presented in the Table 6, 7, and 8.

In Table 6, we compare four traditional face alignment methods, ESR, SDM, LBF and CFSS. It can be seen that our

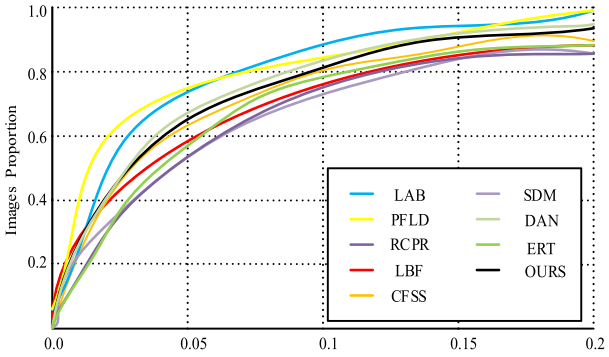

FIGURE 5. CED curves for the 300W dataset.

TABLE 8. Compared test error on AFLW.

Methods	AFLW-Full	AFLW-Frontal	I5	A17
RCPR [26]	3.73	2.87	66	540
ERT [29]	4.35	4.35	15	135
LBF [6]	4.25	2.74	3.1	27
CFSS [8]	3.92	2.68	40	368
TSR [30]	2.17	-	250	2250
LAB [9]	1.85	1.62	1500	-
PFLD [34]	1.88	-	11	87
Ours_res [52]	4.17	2.85	2	21
Ours_res_loss	**3.72**	**2.61**	**2**	**21**

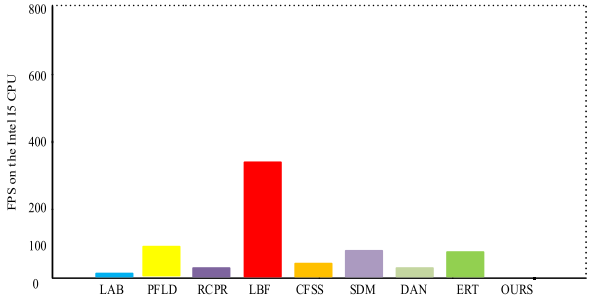

FIGURE 6. Comparison FPS with different methods.

method has obvious advantages in positioning accuracy and running speed compared with ESR, SDM and LBF, the accuracy and running time of our proposed method are improved by 2:17% and 92:66% respectively. Although the accuracy of mean error has reduced by 0.49, the running time of the algorithm is nearly 20 times lower than that of CFSS.

In order to keep consistent with the methods of comparison in the measurement of error, we use two metrics of normalization on the 300W dataset. As can be seen from the tables that the deep learning methods proposed in recent years have significantly improved the positioning accuracy, but it is also very time-consuming. For example, the running time of DAN and LAB reach 150ms and 1500ms on I5 processor respectively. As a contrast, the running time of our proposed method is improved by 3000:1800% and only a small amount of accuracy has been lost, as shown from the Table 6 and 7.

To verify the effectiveness of the new loss shown in Fig. 9, the results of mean error on different losses in Table 6, 7, and 8

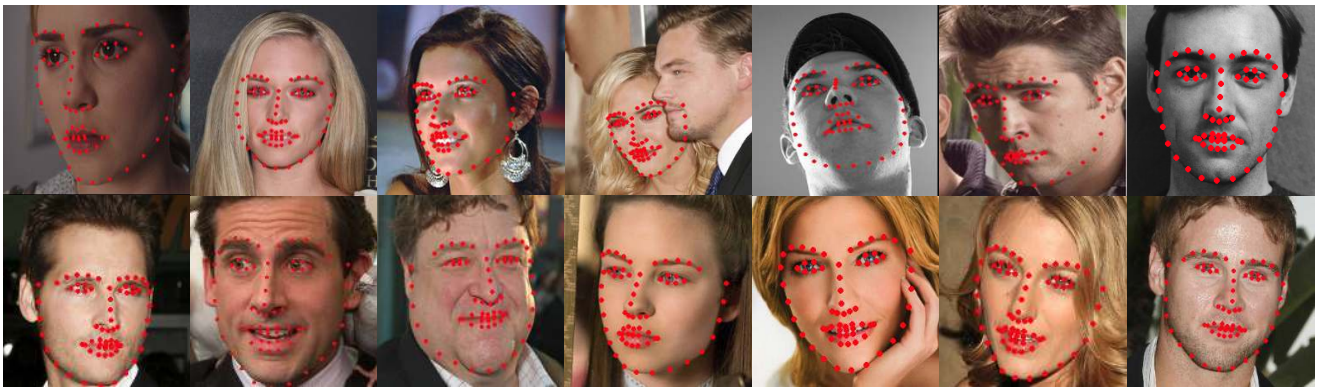

FIGURE 7. Results of predicted landmarks on 300W dataset.

FIGURE 8. Results of predicted landmarks on AFLW datasets.

FIGURE 9. Results of predicted landmarks using different loss (the images at the bottom are trained with the new loss).

are reported. ''Ours_res_loss'' is a model that was trained using the new loss function with two stages network. From the comparison results above, compared with using Euclidean loss, the proposed method model trained with the new loss achieved 4:5.4% improvement in terms of mean error.

Although our approach does not reach state-of-the-art, we have made a great trade-off between running speed and predict accuracy and will be more suitable for deployment and application on mobile platforms.

D. RESULTS

As can be seen from the details of the predicted landmarks in Fig. 9 between two results of the same face image, it solves the problem of inaccurate location of some parts, such as

closed eyes and large angle cheeks. The comparison between ''Ours_res'' [52] and ''Ours_res_loss'' indicates the effectiveness of the new loss compared to the Euclidean loss.

IV. CONCLUSION

Aiming at the shortcomings of the mobile platform, we proposed a real-time, high-precision face alignment method for the mobile platform. The constructed network structure can well learn the location information of key parts, a new loss function has been proposed which can enhance local convergence. The experimental results show that the prediction of new loss in the eyes and cheeks is significantly improved, the overall test mean error has also decreased. Next work, we will pay attention to improving the accuracy of face alignment while maintaining real-time performance. At the same time, we will also try to focus on real-time 3D face alignment.

ACKNOWLEDGMENT

This article was presented in part at the Chinese Conference on Pattern Recognition and Computer Vision. *(Xin Ning and Pengfei Duan contributed equally to this work.)*

REFERENCES

[1] C. Bhagavatula, C. Zhu, K. Luu, and M. Savvides, ''Faster than real-time facial alignment: A 3D spatial transformer network approach in unconstrained poses,'' Jul. 2017, *arXiv:1707.05653*. [Online]. Available: https://arxiv.org/abs/1707.05653

[2] F. Tang, J. Zhang, Y. Feng, Q. Guan, and X. Zhou, ''Real-time face alignment enhancement by tracking,'' in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Qingdao, China, Dec. 2016, pp. 1011–1016.

[3] X. Cao, Y. Wei, F. Wen, and J. Sun, ''Face alignment by explicit shape regression,'' *Int. J. Comput. Vis.*, vol. 107, no. 2, pp. 177–190, Apr. 2014.

[4] X. Xiong and F. De La Torre, ''Supervised descent method and its applications to face alignment,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 532–539.

[5] S. Xiao, J. Feng, J. Xing, H. Lai, S. Yan, and A. Kassim, ''Robust facial landmark detection via recurrent attentive-refinement networks,'' in *Computer Vision*, vol. 9905, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 57–72.

[6] S. Ren, X. Cao, Y. Wei, and J. Sun, ''Face alignment at 3000 FPS via regressing local binary features,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 1685–1692.

[7] D. Lee, H. Park, and C. D. Yoo, ''Face alignment using cascade Gaussian process regression trees,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 4204–4212.

[8] S. Zhu, C. Li, C. C. Loy, and X. Tang, ''Face alignment by coarse-to-fine shape searching,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 4998–5006.

[9] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou, ''Look at boundary: A boundary-aware face alignment algorithm,'' May 2018, *arXiv:1805.10483*. [Online]. Available: https://arxiv.org/abs/1805.10483

[10] M. Kowalski, J. Naruniec, and T. Trzcinski, ''Deep alignment network: A convolutional neural network for robust face alignment,'' Jun. 2017, *arXiv:1706.01789*. [Online]. Available: https://arxiv.org/abs/1706.01789

[11] S. Milborrow and F. Nicolls, ''Locating facial features with an extended active shape model,'' in *Computer Vision*, vol. 5305, D. Forsyth, P. Torr, and A. Zisserman, Eds. Berlin, Germany: Springer, 2008, pp. 504–513.

[12] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, ''300 faces in-the-wild challenge: The first facial landmark localization challenge,'' in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Sydney, NSW, Australia, Dec. 2013, pp. 397–403.

[13] J. Saragih and R. Goecke, ''A nonlinear discriminative approach to AAM fitting,'' in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.

[14] J. M. Saragih, S. Lucey, and J. F. Cohn, ''Deformable model fitting by regularized landmark mean-shift,'' *Int. J. Comput. Vis.*, vol. 91, no. 2, pp. 200–215, Jan. 2011.

[15] S. Yang, P. Luo, C. C. Loy, and X. Tang, ''From facial parts responses to face detection: A deep learning approach,'' Sep. 2015, *arXiv:1509.06451*. [Online]. Available: https://arxiv.org/abs/1509.06451

[16] B. M. Smith and L. Zhang, ''Collaborative facial landmark localization for transferring annotations across datasets,'' in *Computer Vision*, vol. 8694, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 78–93.

[17] Y. Sun, X. Wang, and X. Tang, ''Deep convolutional network cascade for facial point detection,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 3476–3483.

[18] Y. Sun, X. Wang, and X. Tang, ''Deep learning face representation from predicting 10,000 classes,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 1891–1898.

[19] Y. Sun, X. Wang, and X. Tang, ''Hybrid deep learning for face verification,'' in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1489–1496.

[20] G. Trigeorgis, P. Snape, M. A. Nicolaou, E. Antonakos, and S. Zafeiriou, ''Mnemonic descent method: A recurrent process applied for end-to-end face alignment,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 4177–4187.

[21] W. Wu and S. Yang, ''Leveraging intra and inter-dataset variations for robust face alignment,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Honolulu, HI, USA, Jul. 2017, pp. 2096–2105.

[22] Y. Wu, C. Gou, and Q. Ji, ''Simultaneous facial landmark detection, pose and deformation estimation under facial occlusion,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 5719–5728.

[23] S. Zhu, C. Li, C. C. Loy, and X. Tang, ''Unconstrained face alignment via cascaded compositional learning,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 3409–3417.

[24] Y. Liu, A. Jourabloo, W. Ren, and X. Liu, ''Dense face alignment,'' in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Venice, Italy, Oct. 2017, pp. 1619–1628.

[25] Z. Zhu, P. Luo, X. Wang, and X. Tang, ''Deep learning identity-preserving face space,'' in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013.

[26] X. P. Burgos-Artizzu, P. Perona, and P. Dollar, ''Robust face landmark estimation under occlusion,'' in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 1513–1520.

[27] G. Ghiasi and C. C. Fowlkes, ''Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 1899–1906.

[28] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, ''Learning deep representation for face alignment with auxiliary attributes,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 5, pp. 918–930, May 2016.

[29] V. Kazemi and J. Sullivan, ''One millisecond face alignment with an ensemble of regression trees,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 1867–1874.

[30] J. Lv, X. Shao, J. Xing, C. Cheng, and X. Zhou, ''A deep regression architecture with two-stage re-initialization for high performance facial landmark detection,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 3691–3700.

[31] M. Kostinger, P. Wohlhart, P. M. Roth, and H. Bischof, ''Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization,'' in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Barcelona, Spain, Nov. 2011, pp. 2144–2151.

[32] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, ''ShuffleNetV2: Practical guidelines for efficient CNN architecture design,'' Jul. 2018, *arXiv:1807.11164*. [Online]. Available: https://arxiv.org/abs/1807.11164

[33] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, ''MobileNetV2: Inverted residuals and linear bottlenecks,'' Jan. 2018, *arXiv:1801.04381*. [Online]. Available: https://arxiv.org/abs/1801.04381

[34] X. Guo, ''PFLD: A practical facial landmark detector,'' Feb. 2019, *arXiv:1902.10859*. [Online]. Available: https://arxiv.org/abs/1902.10859

[35] X. Dong, Y. Yan, W. Ouyang, and Y. Yang, ''Style aggregated network for facial landmark detection,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 379–388.

[36] S. Wu, J. Xu, S. Zhu, and H. Guo, ''A deep residual convolutional neural network for facial keypoint detection with missing labels,'' *Signal Process.*, vol. 144, pp. 384–391, Mar. 2018.

[37] S. Xiao, S. Yan, and A. A. Kassim, "Facial landmark detection via progressive initialization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Santiago, Chile, Dec. 2015, pp. 986–993.
[38] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, Jun. 2001.
[39] I. Matthews and S. Baker, "Active appearance models revisited," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 135–164, Nov. 2004.
[40] M. Kowalski and J. Naruniec, "Face alignment using k-cluster regression forests with weighted splitting," *IEEE Signal Process. Lett.*, vol. 23, no. 11, pp. 1567–1571, Nov. 2016.
[41] O. Tuzel, T. K. Marks, and S. Tambe, "Robust face alignment using a mixture of invariant experts," vol. 9909, pp. 825–841, 2016, *arXiv:1511.04404*. [Online]. Available: https://arxiv.org/abs/1511.04404
[42] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, "Extensive facial landmark localization with coarse-to-fine convolutional network cascade," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Sydney, NSW, Australia, Dec. 2013, pp. 386–391.
[43] H. Fan and E. Zhou, "Approaching human level facial landmark localization by deep learning," *Image Vis. Comput.*, vol. 47, pp. 27–35, Mar. 2016.
[44] S. Xiao, J. Feng, J. Xing, H. Lai, S. Yan, and A. Kassim, "Robust facial landmark detection via recurrent attentive-refinement networks," in *Computer Vision*, vol. 9905, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 57–72.
[45] A. Bulat and G. Tzimiropoulos, "Two-stage convolutional part heatmap regression for the 1st 3D face alignment in the wild (3DFAW) challenge," vol. 9914, pp. 616–624, 2016, *arXiv:1609.09545*. [Online]. Available: https://arxiv.org/abs/1609.09545
[46] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," Jan. 2018, *arXiv: 1704.08063*. [Online]. Available: https://arxiv.org/abs/1704.08063
[47] J. Wang, C. Lu, M. Wang, P. Li, S. Yan, and X. Hu, "Robust face recognition via adaptive sparse representation," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2368–2378, Dec. 2014.
[48] L. Tran, X. Yin, and X. Liu, "Disentangled representation learning gan for pose-invariant face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1283–1292.
[49] A. Kumar and R. Chellappa, "Disentangling 3D pose in a dendritic CNN for unconstrained 2D Face Alignment," Mar. 2018, *arXiv:1802.06713*. [Online]. Available: https://arxiv.org/abs/1802.06713
[50] D. Cristinacce and T. F. Cootes, "Feature detection and tracking with constrained local models," in *Proc. Brit. Mach. Vis. Conf.*, Edinburgh, U.K., 2006, p. 95.1.
[51] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Robust discriminative response map fitting with constrained local models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 3444–3451.
[52] P. Duan *et al.*, "Faster real-time face alignment method on CPU," in *Proc. Chin. Conf.*, 2019, pp. 398–408.

XIN NING received the Ph.D. degree from the Institute of Semiconductors, Chinese Academy of Sciences, in 2017. He is currently an Assistant Professor of artificial intelligence with the Institute of Semiconductors, Chinese Academy of Sciences (ISCAS). His research interests include deep learning, machine art, pattern recognition, and image cognitive computation.

PENGFEI DUAN received the M.S. degree in aerospace engineering from the Guilin University of Electronic Technology, in 2017. He is currently a Researcher with the Cognitive Computing Technology Joint Laboratory, Wave Group. His research interests include deep learning, pattern recognition, and image cognitive computation.

WEIJUN LI received the Ph.D. degree from the Institute of Semiconductors, Chinese Academy of Sciences, in 2004. He is currently a Professor of artificial intelligence with the Institute of Semiconductors, Chinese Academy of Sciences (ISCAS), and the University of Chinese Academy of Sciences. He is in charge of the Artificial Intelligence Research Center of ISCAS, and also the Director of the Laboratory of High Speed Circuits and Neural Networks of ISCAS. His research interests include deep modeling, machine art, pattern recognition, artificial neural networks, and intelligent systems.

YUAN SHI received the M.S. degree from the Wuhan University of Technology, in 2014. He is currently a Researcher with the Cognitive Computing Technology Joint Laboratory, Wave Group. His research interests include deep learning, pattern recognition, and image cognitive computation.

SHUANG LI received the B.Eng. degree from the Beijing University of Chemical Technology, in 2017. He is currently a Research Assistant of artificial intelligence with the Institute of Semiconductors, Chinese Academy of Sciences (ISCAS). His research interests include deep learning, pattern recognition, and image cognitive computation.