# PicassoNet: Searching Adaptive Architecture for Efficient Facial Landmark Localization

Tiancheng Wen, Zhonggan Ding, Yongqiang Yao, Yaxiong Wang, and Xueming Qian, *Member, IEEE*

***Abstract*— Since recent facial landmark localization methods achieve satisfying accuracy, few of them enable fast inference speed, which, however, is critical in many real-world facial applications. Existing methods typically employ complicated network structure and predict all the key points through uniform computation, which is inefficient since individual facial part might take different computation to obtain the best performance. Taking both accuracy and efficiency into consideration, we propose the PicassoNet, a lightweight cascaded facial landmark detector with adaptive computation for individual facial part. Different from the conventional cascaded methods, PicassoNet integrates refinement submodules into a single network with group convolution, where each convolution group predicts landmarks from an individual facial part. Note that the groups' structures are flexible in the training process. Then, a novel grouping search algorithm is proposed to optimize the group division. With formulating the optimization as a network architecture search (NAS) problem, the grouping search adaptively allocates computation to each group and obtains an efficient structure. In addition, we propose a boundary-aware loss to optimize along tangent and normal of facial boundaries, instead of optimizing along horizontal and vertical as the conventional loss (L2, SmoothL1, WingLoss, and so on) do. The novel loss improves the joint locations of predicted keypoints. Experiments on three benchmark datasets AFLW, 300W, and WFLW show that the proposed method runs over 6× times faster than the state of the arts and meanwhile achieves comparable accuracy.**

***Index Terms*— Facial landmark localization, learnable group convolution, network architecture search (NAS).**

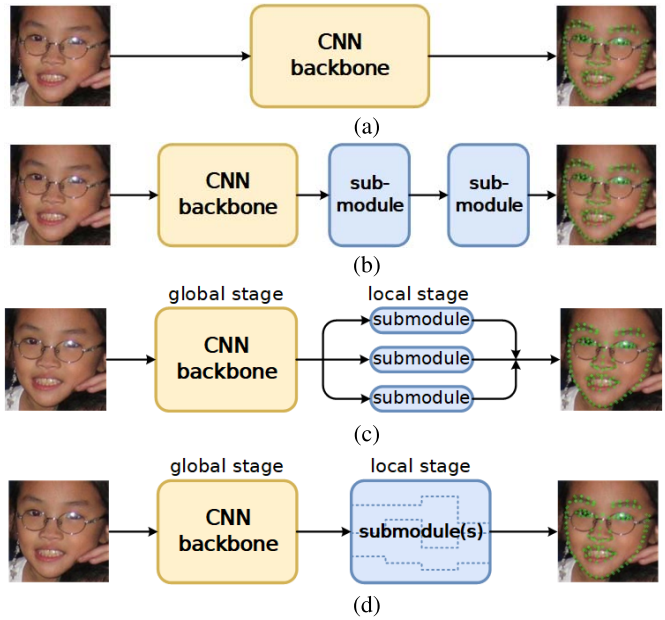



Fig. 1. Pipelines of the existing methods and ours. Different from existing frameworks, we propose to utilize the group convolution to integrate the refinement submodules into a single network, aiming at achieving adaptive computation for different facial parts. (a) Single network pipeline. (b) Pipeline with serial submodules. (c) Pipeline with parallel submodules. (d) Our pipeline with single local network composed of structurally adaptive submodules. Each submodule is marked as the area enclosed by the dotted lines.

## I. Introduction

FACIAL landmark localization aims to predict the coordinates of predefined key points for face images, which plays a critical role in various face analysis tasks, such as face recognition [1], [2], face manipulation [3], [4], and 3-D face reconstruction [5], [6]. As a prerequisite component, landmark localization is requested to achieve not only satisfactory accuracy but also high run-time efficiency, especially in mobile applications. However, most state-of-the-art algorithms are time-consuming, which need enormous parameters and run in low efficiency, leading to difficulties in practical deployment.

The overall pipeline of the existing face alignment methods can be roughly divided into three categories: single network [2], [7], [8], cascaded network with serial identity submodules [9], [10], and cascaded network with parallel identity submodules [11]–[13], as shown in Fig. 1. The predicted landmarks are computed through uniform structure in these pipelines. Nevertheless, different regional areas of face image might differ in requisite computation to achieve decent accuracy because the number of annotated landmarks and the shape variation of individual part are quite different. For instance, human mouth has abundant poses in various emotions and languages, while human nose is nearly rigid. Therefore, identity computing structure and complexity are inflexible and uneconomic for the topic.

Manuscript received March 4, 2021; revised January 29, 2022; accepted April 1, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0101501 and in part by the Science and Technology Program of Xi'an, China, under Grant 21RGZN0017. *(Corresponding author: Xueming Qian.)*

Tiancheng Wen is with the School of Information and Communications Engineering and the Smiles Laboratory, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China (e-mail: ssstormix@stu.xjtu.edu.cn).

Zhonggan Ding and Yongqiang Yao are with the YouTu Lab, Tencent, Shanghai 200030, China (e-mail: 1325374556@qq.com; kelvinyao@tencent.com).

Yaxiong Wang and Xueming Qian are with the Ministry of Education Key Laboratory for Intelligent Networks and Network Security, School of Information and Communications Engineering, and the Smiles Laboratory, Xi'an Jiaotong University, Xi'an, Shannxi 710049, China (e-mail: qianxm@mail.xjtu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3167743.

Digital Object Identifier 10.1109/TNNLS.2022.3167743

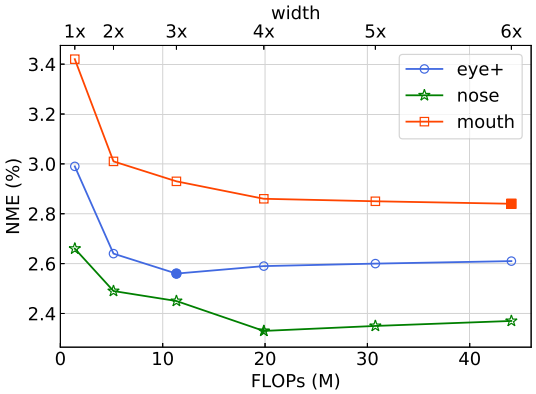



Fig. 2. NME variation along model FLOPs. A metamodel with five convolutional layers is trained on 300W, whose FLOPs increases by multiplying channels. "eye+" is the composite area of eye and eyebrow. The optimal points (filled markers) cost different FLOPs.

To better analyze the computation on individual area, we perform a toy experiment about localizing the landmarks in different facial regions. Given specific facial part (eye/nose/mouth), models with different widths are trained and evaluated. The results shown in Fig. 2 indicate that three facial parts take different floating-point operations per second (FLOPs) to achieve the optimal performance. Therefore, we could assume that: 1) it is a simple task to localize landmarks for some specific local facial areas and, therefore, it is unnecessary to take enormous computations at local refinement stage and 2) submodules of individual facial parts differ in requisite computation complexity. Surplus FLOPs might even cause performance degradation.

Motivated by the above observations, we present a speed–accuracy balanced method, which constructs adaptive computation for each facial part. A global–local cascaded network in parallel manner is employed as the backbone, as shown in Fig. 1(d). In particular, the global network is fed with an input face image and outputs the initial estimations of all landmarks. In the local refinement stage, different facial parts are cropped and recomposed into a single tensor as the input of the local network, which is a single compact network that applies group convolution to decompose the input and handle different facial parts, respectively. This design enables the proposed local network to flexibly allocate computations to each facial part by adjusting the grouping hyperparameters in each convolution layer. Furthermore, to determine the optimal hyperparameters in group convolutions, we introduce a novel grouping search algorithm, which combines the differentiable architecture search and parameterized group convolution. We note the proposed framework as PicassoNet since the recomposition/decomposition manner is sparked by Pablo Picasso's cubism portraitures [such as Seated Woman (1937) and Portrait of Sabartes (1939)], where facial features are deconstructed, analyzed, and reassembled.

In addition, we propose decomposing loss (D-Loss), a novel boundary-aware object function to solve the problem that connecting lines of the predicted landmarks do not always fit the facial boundaries well. Different from conventional regression losses (L2, SmoothL1 [14], WingLoss [15], and so on) that measure errors along the horizontal and vertical directions, the proposed object function optimizes the loss along the tangent and normal directions of the facial boundaries and rebalances the contributions from these two directions.

In summary, the main contributions of this article are given as follows.

1) We present PicassoNet, a region-based cascaded network for real-time facial landmark localization. PicassoNet flexibly allocates the computations to different facial parts in the local refinement stage and thus obtains decent accuracy with low computation complexity.
2) We propose a grouping search algorithm to optimize the structure for PicassoNet. The algorithm helps to find the optimal grouping for each layer in a differentiable manner, which is equivalent to adaptively allocating computations to each facial part.
3) We propose D-Loss, a novel regression object function that measures the loss along the tangent and normal directions of facial boundaries.
4) On mainstream datasets including AFLW [16], 300W [17], and WFLW [7], the proposed PicassoNet achieves comparable performance and runs over 6× times faster than the state of the arts, e.g., it takes about 24.1 ms per image on CPU (i7-8700 at 3.20 GHz). When implemented with specialized mobile framework, it only needs 7.1 ms on arm (Apple A10 Fusion at 1.30 GHz).

## II. Related Works

### A. 2-D Facial Landmark Localization

Recent developments on facial landmark localization mainly focus on designing network structure or objective function and introducing extra data or annotations. The overall structure of the existing methods can be roughly divided into three categories: single network, cascaded network with serial submodules, and cascaded network with parallel submodules. As for the single network pipeline, Wu *et al.* [7], Bulat and Tzimiropoulos [18], and Yang *et al.* [19] employed a stacked hourglass network, which was at first proposed to solve the human pose estimation problem [20]. Bulat and Tzimiropoulos [18] proposed FAN, predicting facial landmarks with hourglass structure for the first time. Yang *et al.* [19] introduced supervised face transformation before CNN backbone to align the input face images. Zhu *et al.* [21] appended three close-knit CNN modules after the backbone to solve the occlusion problem. Also, Wang *et al.* [22] proposed HRNet to exploit multiresolution representation to locate landmarks. These methods, roughly sketched in Fig. 1(a), generally yield satisfying accuracy, but enormous parameters and large feature map resolution seriously affect the run-time speed in practical applications.

There are also many works employing cascaded networks, some of which arrange submodules in a serial way, as shown in Fig. 1(b). Typically, Kowalski *et al.* [9] and Dapogny *et al.* [10] stacked replicated networks to iteratively refine the landmark locations, and Merget *et al.* [23] designed a global–local context network with kernel convolution and

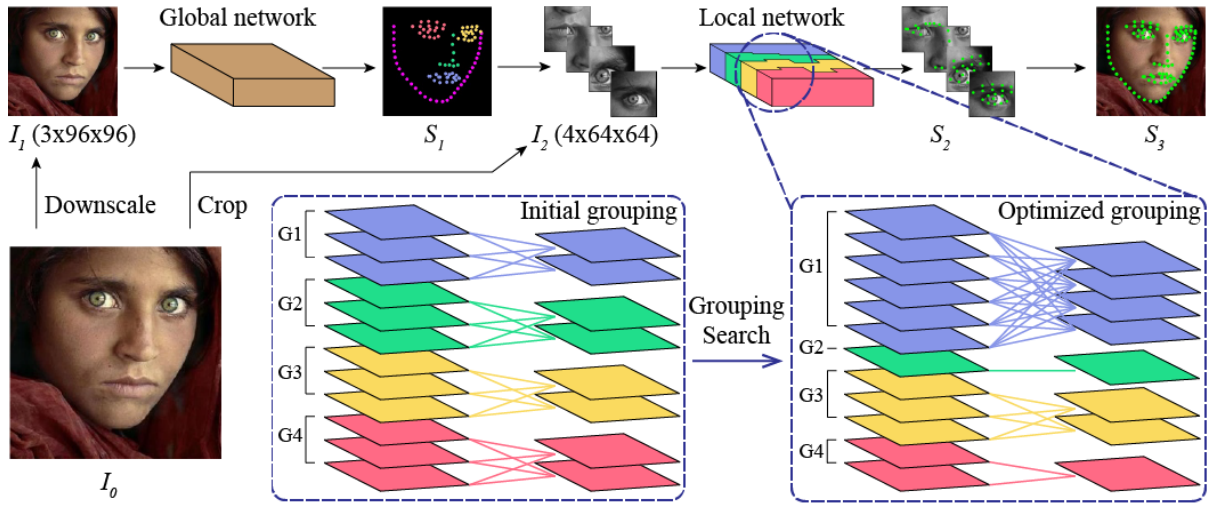



Fig. 3. Framework of the proposed PicassoNet. The global network makes coarse predictions at first. Cropped images are concatenated along the channel dimension as a single tensor. The local network applies four groups to decompose the compound input, marked as G1, G2, G3, and G4. Also, grouping search helps to find the optimal grouping division. Note that each layer might differ in grouping state, and therefore, the local network looks like a Tetris block assemble in overall appearance.

dilated convolution to exploit global information. Besides, it is also effective to arrange the submodules in a parallel way, as shown in Fig. 1(c), whose global–local architecture can capture more details with restricted input resolution. Sun *et al.* [24] and Zhou *et al.* [25] fed each subnetwork with the corresponding input regions. Chandran *et al.* [13] also designed a global–local cascaded pipeline in a heatmap regression and end-to-end manner. Note that the submodules in cascaded networks share the same structure, which means that it takes equal computations to predict individual facial landmark. As a result, identity submodule is detrimental to improving accuracy and economizing FLOPs.

Object function in facial landmark localization has also received attention recently. Feng *et al.* [15] proposed WingLoss to focus on small regression errors, Wang *et al.* [26] designed adaptive WingLoss to balance loss contribution from foreground and background, and Lai *et al.* [27] proposed a normalized mean error (NME) loss to directly optimize the NME. These objective functions optimize the training loss from the horizontal and vertical directions, namely, $x$- and $y$-axis directions, which might lead to the phenomenon that some predicted keypoints shift jointly, though the numerical value of the conventional loss is tiny.

Some works have proved that extra data or annotation contributes to improving the landmark accuracy. For instance, Dong *et al.* [28] introduced a generative adverbial network (GAN) to augment the dataset, and Dong and Yang [29] employed a teacher–student framework and utilized pseudo annotations to improve the performance. With the help of landmark visibility annotations, Kumar *et al.* [8] investigated to model the uncertainty and visibility learning to improve the performance.

### B. Learnable Group Convolution

Group convolution has shown its capability of accelerating deep models in recent efficient networks, such as MobileNet [30], [31] and ShuffleNet [32], [33]. Beyond the conventional even and fixed partition, some recent works investigate learnable grouping. FLGC [34] parameterizes group convolution and formulates a differentiable optimization, but the training is unstable and severely sensitive to initialization, because its softmax relaxation shields the non-maximum paths and extremely shrinks the search space. Zhang *et al.* [35] denoted the group convolution with Kronecker product, while whose representation space contains many duplicate instances and lacks some specific instances. For instance, a square matrix whose number of rows (columns) is prime cannot be decomposed via the Kronecker product. In this work, the proposed grouping search method tackles the existing problems, formulating a complete search space and obtaining a better group partition. Besides, each group of PicassoNet is responsible for a specific facial region, which has a precise physical meaning. The discussed works, including this work, treat learnable grouping as a continuous architecture search process according to [36].

## III. Methodology

### A. System Overview

The proposed PicassoNet consists of a global network and a local network, as shown in Fig. 3. Given a downscaled input face image $I_1$, the global network holistically predicts the coordinate of all landmarks, which is denoted as $S_1$. Contour points of $S_1$ go directly as the final output, while other points instruct to crop local area from the original high-resolution image $I_0$. The cropped regions capture more details than $I_1$ while still in a low resolution. With this design, globally localizing landmarks is decomposed into regional regression problems. Besides, $S_1$ can help to tackle large poses and exaggerated expressions, which further eases the task.

In respect of the local refinement stage, we crop out four facial parts according to $S_1$, namely, left eye, right eye, nose, and mouth. Note that the eye area covers the eyebrow. The cropped facial parts are converted to grayscale and

concatenated along the channel dimension, composing input tensor $I_2$ in Fig. 3. Fed with the reconstructed input tensor, a single local network rather than conventional replicated subnetworks conducts local regression on these facial parts. In particular, the local network adopts group computation for convolution layer and linear layer. Therefore, regional regression is isolated with each other. We further design a grouping search algorithm to optimize the grouping state for each layer, which can be viewed as an architecture search process. The final outputs combine contour points from $S_1$ and local area points from $S_2$. Note that both global network and local network use extremely downscaled inputs and coordinate regression project head, which runs in high efficiency but generally performs weakly. While with the proposed grouping search algorithm exploiting network capacity, the lightweight design is able to obtain decent accuracy compared with existing complicated models.

### B. Generalizing Regional Submodules

As observed from Fig. 2, different facial parts might differ in requisite computation complexity to localize local landmarks. This is because each part has its own morphological characteristics and annotations. For instance, nose usually maintains a rigid shape, while lips vary a lot in appearance on account of expression and makeup. To avoid tedious work on customizing individual network for each facial part, we aim to find an efficient and effective way. In this section, we adopt group convolution to generalize replicated subnetworks of conventional global–local cascaded methods, in order to formulate a representation for joint optimization.

Considering a local network whose layers are equally grouped, it can be viewed as an integration of replicated submodules, where a submodule is determined by each layer's filters that belong to the same group, as shown in "initial grouping" in Fig. 3. To be more specific, when all the layers divide the input tensor and filters into equally sized groups, the local network turns to conventional parallel submodules that share the same architecture. When the partition is unbalanced, the local network allocates different computation complexity to each facial part. Furthermore, when each layer's grouping state differs from each other, the local network equals a set of heterogeneous submodules.

Dividing input tensor and filters into several nonoverlapping groups, group convolution conducts computation inside each group and outputs the concatenated results. Mathematically, for $k$th convolution layer who has $G^k$ groups, the filter weights $W^k$ and input $X^k$ are defined as (1) and (2), where $\cup$ notes the concatenating along channel and the subscript is group index. The output is calculated as (3), where $\otimes$ is the regular convolution between convolutional filter and input tensor

$$W^k = W_1^k \cup W_2^k \cup \cdots \cup W_{G^k}^k \tag{1}$$

$$X^k = X_1^k \cup X_2^k \cup \cdots \cup X_{G^k}^k \tag{2}$$

$$Y^k = \left(W_1^k \otimes X_1^k\right) \cup \left(W_2^k \otimes X_2^k\right) \cup \cdots \cup \left(W_{G^k}^k \otimes X_{G^k}^k\right). \tag{3}$$

In PicassoNet, we set $G^k = 4$ to localize landmarks on the left eye, right eye, nose, and mouth. A critical issue emerges: how to find the optimal grouping state for each layer? As a matter of fact, the local network with the above generalization can be further regarded as a supernet that contains various subnetworks. Hence, how to find the optimal grouping state is transformed into a network architecture search (NAS) problem, where layers' grouping states compose the search space. We then propose a differentiable grouping search algorithm to solve the problem, which will be discussed in Section III-C.

### C. Grouping Search

We at first interpret the grouping state in mathematics and then explore an automated search method. For convenience, we refer to the parameterization of FLGC [34], which parameterizes the $k$th layer's grouping state with two binary mask matrices $M_{\text{in}}^k$ and $M_{\text{out}}^k$ in shapes of $C_{\text{in}}^k \times G^k$ and $C_{\text{out}}^k \times G^k$, where $C_{\text{in}}^k$ and $C_{\text{out}}^k$ denote the $k$th layer's input channel and output channel, respectively. The definitions of $M_{\text{in}}^k$ and $M_{\text{out}}^k$ are formulated as follows:

$$M_{\text{in}}^k(i,j) = \begin{cases} 1, & \text{if } x_i^k \in X_j^k \\ 0, & \text{if } x_i^k \notin X_j^k \end{cases} \quad i \in \left[1, C_{\text{in}}^k\right]; \quad j \in \left[1, G^k\right] \tag{4}$$

$$M_{\text{out}}^k(i,j) = \begin{cases} 1, & \text{if } w_i^k \in W_j^k \\ 0, & \text{if } w_i^k \notin W_j^k \end{cases} \quad i \in \left[1, C_{\text{out}}^k\right]; \quad j \in \left[1, G^k\right] \tag{5}$$

where $x_i^k$ represents the $i$th channel of $X^k$ and $w_i^k$ is the $i$th filter. When $x_i^k$ or $w_i^k$ is divided into the $j$th group, the matrix value in the corresponding position ($M_{\text{in}}^k(i,j)$ or $M_{\text{out}}^k(i,j)$) is activated to 1 and otherwise 0.

In PicassoNet, $M_{\text{out}}^k$ literally equals $M_{\text{in}}^{k+1}$. This is because local regressions are isolated from each other, which indicates that each layer's output channel shares an identical division with its succeeding layer's input channel; otherwise, features from disparate facial parts might confuse the prediction. Therefore, the group convolution in (3) can be rederived as (6), where $\odot$ is the Hadamard product and $M^k$ is the $k$th layer's sole parameter to be optimized. In particular, a convolutional filter in the $j$th group who is responsible for the $i$th output channel, modified with the $i$th column of $M^{k+1}(M^k)^T$, reserves elements whose corresponding input channels belong to the $j$th group, while it has other elements shielded. Therefore, all the possible values of $M^k$ compose the topology structure space of the $k$th layer, and combinations of layers with any possible structure compose the complete search space

$$\begin{aligned} Y^k &= \left(W^k \odot \left(M_{\text{out}}^k \left(M_{\text{in}}^k\right)^T\right)\right) \otimes X^k \\ &= \left(W^k \odot \left(M_{\text{in}}^{k+1} \left(M_{\text{in}}^k\right)^T\right)\right) \otimes X^k \\ &= \left(W^k \odot \left(M^{k+1} \left(M^k\right)^T\right)\right) \otimes X^k. \end{aligned} \tag{6}$$

Seeking the optimal grouping state for the $k$th layer is equivalent to optimize $M^k$. However, the binary matrix is not differentiable. As a solution, we optimize a contiguous substitution $\overline{M}^k$ in the same shape and use the Gumbel softmax [37] to map $\overline{M}^k$ to $\hat{M}^k$ within binary space as the approximation of $M^k$. Note that the softmax relaxation in [34]

is deprecated because softmax-based search might be sensitive to the initialization of mask matrices, which will be discussed in the following analysis. The relaxation is described in (7), where $g(i, j) \sim \text{Gumbel}(0, 1)$ is a random noise subject to the Gumbel distribution. Also, approximation $\hat{M}^k(i, j)$ following continuous distribution could be updated via backpropagation, and thus, it is easy to be embedded in network training:

$$\begin{aligned}\hat{M}^k(i, j) &= \text{GSoftmax}\Big(\overline{M}^k(i, j)\Big) \\ &= \frac{\exp\Big(\overline{M}^k(i, j) + g(i, j)\Big)}{\sum_j \exp\Big(\overline{M}^k(i, j) + g(i, j)\Big)}.\end{aligned} \tag{7}$$

*Why Not Softmax?* After the softmax operation, the value $\text{Softmax}(\overline{M}^k(i, j))$ represents the probability that channel $i$ belongs to group $j$. Then, the forward computation selects the group for each channel according to where the maximum probability appears. However, the maximum position of $\text{Softmax}(\overline{M}^k(i, ))$ exactly locates at the maximum position of $\overline{M}^k(i, )$, which actually eliminates the randomness of probability. In other words, for channel $i$, nonmaximum group choices could hardly be activated and updated, especially when mask matrices are initialized improperly and network weights are pretrained deficiently. This nonrandom selection limits the layer to switch into a very different grouping state and thus extremely shrinks the search space. On the contrary, Gumbel softmax introduces Gumbel noise to the selection and practically conducts sampling according to $\overline{M}^k(i, j)$, which means that the maximum position of $\text{GSoftmax}(\overline{M}^k(i, j))$ might not locate at the maximum position of $\overline{M}^k(i, j)$. Therefore, all the elements of $\overline{M}^k(i, j)$ have a chance to be picked and updated even though some are small. Also, we can simply initialize the mask matrices as the all-ones matrix.

Directly optimizing $\overline{M}^k$ along with $W^k$ is unstable because the weight space might vary a lot as the network architecture changes. It is essential to fix the architecture when solving weights. Therefore, we formulate the optimizing object following series works of DARTS [35], as shown in (8), where $\mathcal{L}_{\text{train}}$ and $\mathcal{L}_{\text{val}}$ are the training and the validation loss that take turns to be updated, respectively:

$$\begin{aligned}&\min_{\hat{M}} \mathcal{L}_{\text{val}}\big(\hat{M}\big|W\big) \\ &\min_{W} \mathcal{L}_{\text{train}}\big(W\big|\hat{M}\big).\end{aligned} \tag{8}$$

In practical implementation, only network weights are updated during the first $N$ iterations to warm up the group choices for each channel. Then, $\mathcal{L}_{\text{train}}$ and $\mathcal{L}_{\text{val}}$ are optimized alternately. The overall optimizing procedure is summarized in Algorithm 1, where $\nabla$ is the derivation and $J$ is the all-ones matrix. $\hat{M}^k$ activates a topology path that connects the input channels and the output channels before forward computation.

**Algorithm 1** Solving the Optimization Problem in (8)

**Input:** $X_{train}$: data from training dataset; $X_{val}$: data from validation dataset;
**Output:** $\overline{M}^k, W^k : k \in [1, K]$;

```
Initialize W^k ← msra[1]; M̄^k ← J;
for i = 0 to N do
    M̂^k ← GSoftmax(M̄^k);
    Update W^k by descending ∇_W L_train(W|M̄) for a batch of X_train;
end for
while not converged do
    M̂^k ← GSoftmax(M̄^k);
    Update M̄^k by descending ∇_M̂ L_val(M̂|W) for a batch of X_val;
    M̂^k ← GSoftmax(M̄^k);
    Update W^k by descending ∇_W L_train(W|M̂) for a batch of X_train;
end while
return M̄^k, W^k
```

### D. Decomposing Loss

In this section, we introduce the D-Loss. A common phenomenon in facial landmark localization is that some predicted key points do not fit the physical boundaries well, which can be observed in Fig. 4. Also, we investigate the problem from the view of loss function's penalizing directions. The conventional coordinate regression loss functions (L1, L2, SmoothL1 [14], Wing [15], and so on) calculate the distance along the $x$- and $y$-axes, which might lead to the phenomenon that some predicted landmarks shift jointly, though their numerical values of the conventional loss are tiny. While taking the above issue into consideration, we attempt to penalize the training loss in the tangential direction and normal direction of boundaries (orange lines in Fig. 4). The loss can be formulated as

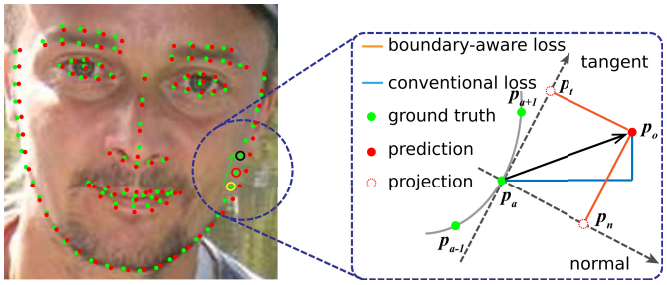



Fig. 4. Problem that predictions do not fit the boundary. D-Loss aims to penalize the distance along the tangential and normal directions rather than the horizontal and vertical directions. For specific point (e.g., red circled), the tangential direction is approximated with the ray from the prior point (yellow circled) to the succeeding point (black circled).

$$\mathcal{L}_{D\text{-Loss}} = \alpha\mathcal{F}(p_t, p_a) + (2 - \alpha)\mathcal{F}(p_n, p_a) \tag{9}$$

where $p_a$ is the ground-truth position of given landmark. $p_t$ and $p_n$ are projected positions of prediction $p_o$ in tangent and normal of the facial boundary, respectively. $\mathcal{F}$ notes the distance criterion such as SmoothL1. Also, $\alpha$ is the weight to balance the contribution from two directions. When $\alpha$ equals 1, D-Loss transforms into conventional $\mathcal{F}(p_o, p_a)$. To solve the boundary fit problem, $\alpha$ is supposed to be set smaller than 1 to make the model pay more attention to optimize the normal component. In practice, the tangent direction is approximated

by connecting $p_a$'s prior point and succeeding point in ground truth.

*D-Loss Versus LUVLi:* Kumar *et al.* [8] also proposed to decompose the loss along different directions, and however, it has considerable differences with our D-Loss. First, LUVLi attempts to locate the landmarks by directly computing the weighted sum of the activated heatmap, and such a manner would introduce uncertainty from all directions. While our D-Loss only focuses on the tangential and normal directions. Second, LUVLi requires the heatmap to perform the estimation, while the heatmap is not a requisite for our PicassoNet. Finally, LUVLi pays more attention to the tangential errors, while we argue that the smaller normal errors have more impact, since the label on the normal direction is more reliable due to human's higher sensitive to the normal direction.

## IV. Experiments

### A. Experimental Setup

*1) Datasets:* We evaluate the proposed PicassoNet on popular facial landmark datasets, including 300W [17], WFLW [7], and AFLW [16]. The numbers of facial landmarks of these datasets are 68, 98, and 21, respectively; 300W contains 3148 training samples and 689 test samples, which is further divided into 135 challenge samples and 554 common samples. WFLW collects 7500 training faces and 2500 test faces from WIDER Face [39], and its variations in expression, pose, and occlusion bring many difficulties to existing approaches. AFLW provides 24 386 faces in total. Following previous works, we use 20 000 training samples and 4386/1314 test samples as a full/frontal set.

*2) Evaluation Metrics:* To align with most of the existing methods, we compute the NME on each dataset, which is defined as

$$\text{NME}(p_i, \hat{p}_i) = \frac{1}{N} \sum_{i=1}^{N} \frac{\|p_i - \hat{p}_i\|^2}{d} \tag{10}$$

where $p_i$ and $\hat{p}_i$ denote the ground truth and the predicted coordinate, respectively, $N$ is the number of landmarks, and $d$ represents the normalization distance, which is Inter-Ocular-Norm (ION) on 300W and WFLW, while for AFLW, $d$ goes to the width of face bounding box. Further statistics are reported for comprehensive analysis, including failure rate (FR) and area under curve (AUC) based on cumulative error distribution (CED). We set the threshold as 0.1 to calculate these scores. Besides, we take FLOPs to evaluate efficiency, which is positively related to practical processing speed on CPU, which is the mainstream scenario for mobile face alignment deployments.

*3) Implementation Details:* To verify the effectiveness of the PicassoNet under extremely limited computations, we build the global network and the regional network based on the inverted residual block (IRB) [31] and downsample the input image to $96 \times 96$ and $64 \times 64$ for the global network and local network. The overall computations occupy about 105.69 MFLOPs, which is about one-tenth of state of the arts. The detailed structures of global network and local network are reported in Tables I and II. The local network has four convolution groups for left eye, right eye, nose, and mouth ($G_k = 4$). Note that the partition depends on the natural human face appearance. Any other partition, such as merging the two eye areas or making more divisions, will produce computationally inefficient rectangle area rather than nearly square area or break the natural face appearance.

TABLE I

Global Network Configuration. $t, c, n$, and $s$ Represent IRB's Expanding Ratio, Output Channel, Number of Layers, and Stride, Respectively. Multiscale Features Are Extracted and Concatenated to Obtain the Final Prediction

| input | operator | $t$ | $c$ | $n$ | $s$ | output |
|---|---|---|---|---|---|---|
| $96^2 \times 3$ | Conv3x3 | - | 24 | 1 | 1 | $96^2 \times 24$ |
| $96^2 \times 24$ | IRB | 1 | 24 | 1 | 2 | $48^2 \times 24$ |
| $48^2 \times 24$ | IRB | 5 | 32 | 2 | 2 | $24^2 \times 32$ |
| $24^2 \times 32$ | IRB | 5 | 64 | 5 | 2 | $12^2 \times 64$ |
| $12^2 \times 64$ | IRB | 5 | 128 | 6 | 2 | $6^2 \times 128$ ($F_0$) |
| $F_0$ | Conv1x1 | - | 64 | 1 | 1 | $6^2 \times 64$ ($F_1$) |
| $F_0$ | Avgpool | - | - | 1 | - | $3^2 \times 128$ |
| $3^2 \times 128$ | Conv1x1 | - | 128 | 1 | 1 | $3^2 \times 128$ ($F_2$) |
| $F_0$ | Avgpool | - | - | 1 | - | $1^2 \times 256$ |
| $1^2 \times 256$ | Conv1x1 | - | 256 | 1 | 1 | $1^2 \times 256$ ($F_3$) |
| $F_1, F_2, F_3$ | Linear | - | 136 | 1 | - | $1 \times 136$ |

TABLE II

Local Network Configuration. Parameter $g$ Notes the Number of Groups for Each Layer, and the Other Parameters Are the Same as Those in Table I

| input | operator | $t$ | $c$ | $n$ | $s$ | $g$ | output |
|---|---|---|---|---|---|---|---|
| $64^2 \times 4$ | Conv3x3 | - | 24 | 1 | 1 | 4 | $64^2 \times 24$ |
| $64^2 \times 24$ | IRB | 1 | 24 | 1 | 2 | 4 | $32^2 \times 24$ |
| $32^2 \times 24$ | IRB | 5 | 24 | 2 | 2 | 4 | $16^2 \times 24$ |
| $16^2 \times 24$ | IRB | 5 | 56 | 5 | 2 | 4 | $8^2 \times 56$ |
| $8^2 \times 56$ | IRB | 5 | 104 | 6 | 2 | 4 | $4^2 \times 104$ |
| $4^2 \times 104$ | Conv1x1 | - | 248 | 1 | 1 | 4 | $4^2 \times 248$ |
| $1 \times 3968$ | Linear | - | 136 | 1 | - | 4 | $1 \times 136$ |

We use PyTorch [40] to conduct all the experiments. Torchstat[2] is applied to analyze computations. Translation, rotation, scale, and random flip are conducted as data augmentation during training. We train the global network using the Adam optimizer with an initial learning rate of $5 \times 10^{-3}$ for 150 epochs. The training of regional network includes the search stage and fine-tuning stage, which is similar to DARTS [41]. For the search stage, we by turns update grouping mask and network weights for 500 epochs. Then, the fine-tuning stage uses the same settings as global network's training. As for the loss function, WingLoss is applied as $\mathcal{F}$ and $\alpha$ of D-Loss is set as 0.6.

### B. Comparison With Existing Approaches

*1) Evaluation on 300W:* For experiments on 300W, we rebalance the training samples according to pose coefficients to tackle extreme pose variations, which is similar to the PDB process [15]. For global network training, the random scale range of images is [1.1, 1.16] since dataset 300W has less profile keypoints; therefore, we set relatively larger

[2]https://github.com/Swall0w/torchstat

TABLE III
NME (%) ON 300W COMMON SET, CHALLENGE SET, AND FULLSET

| Method | Year | Com. | Chall. | Full | FLOPs (G) |
|---|---|---|---|---|---|
| MDM [47] | 2016 | 4.83 | 10.14 | 5.88 | $-^{\dagger}$ |
| DAN [12] | 2017 | 4.42 | 7.57 | 5.03 | 3.76 |
| AAN [48] | 2018 | 4.38 | 9.44 | 5.39 | 0.939 |
| CPM+SBR [49] | 2018 | 3.28 | 7.58 | 4.10 | 21.1 |
| DSRN [50] | 2018 | 4.12 | 9.68 | 5.21 | 3.29 |
| LAB [7] | 2018 | 2.98 | 5.19 | 3.49 | 19.0 |
| SAN [28] | 2018 | 3.34 | 6.60 | 3.98 | 21.1 |
| CU-Net [51] | 2019 | 3.19 | 5.76 | 3.69 | 2.87 |
| TS3 [29] | 2019 | 3.17 | 6.41 | 3.78 | 3.33 |
| ODN [21] | 2019 | 3.56 | 6.67 | 4.17 | 1.58 |
| Laplace KL [52] | 2019 | 3.19 | 6.87 | 3.91 | 2.99 |
| AWing [26] | 2019 | 2.72 | **4.52** | **3.07** | 8.73 |
| 3DDE [53] | 2019 | **2.69** | 4.92 | 3.13 | $1.12^{\ddagger}$ |
| HRNet [22] | 2020 | 2.91 | 5.11 | 3.34 | 4.71 |
| 3FabRec [45] | 2020 | 3.36 | 5.74 | 3.82 | 2.20 |
| ADC [13] | 2020 | 2.83 | 7.04 | 4.23 | 26.2 |
| **Global stage (Ours)** | - | 3.38 | 6.47 | 3.99 | 0.097 |
| **PicassoNet (Ours)** | - | 3.03 | 5.81 | 3.58 | **0.106** |

$^{\dagger}$ MDM [47] doesn't report the exact network architecture.
$^{\ddagger}$ FLOPs of 1.12G includes CNN and excludes regression trees.

scale to harvest more accurate localizations on eye/nose/mouth keypoints. The images are randomly rotated within $\pm 16^{\circ}$ and translated within $\pm 7$ pixels horizontally and vertically. For the local network training, the scale range, rotation range, and translation range are [0.9, 1.0], [-6°, 6°], and $[-3, 3]$ pixels, respectively. NME results are reported in Table III, where we compare the PicassoNet with state-of-the-art methods. With the lowest computation, the proposed method outperforms the majority of comparative methods [12], [13], [21], [28], [29], [45], [48]–[52], and underperforms LAB [7], AWing [26], and HRNet [22], which costs much more computations. In particular, FLOPs of 0.106G are lower by more than an order of magnitude among existing approaches. The tiny cost comes from low-resolution input and concise architecture design (Literally ResNet18 with $224 \times 224$ input takes up to 1.82 GFLOPs).

Just as the global network behaves, compact network without any specialized design could not make precise predictions. However, the scores go far more accurate when refined by the local network, which brings little increase on FLOPs (nearly 10%). The phenomenon verifies the first assumption we proposed in Section I. It is a simple task to localize landmarks on a local facial area, and therefore, enormous computation cost of refinement stage is unnecessary.

*2) Evaluation on WFLW:* WFLW's training dataset covers many faces in large poses, and therefore, the pose balancing is not conducted on WFLW. Due to the diversity of dataset WFLW, images of WFLW are not scaled for both global network and local network. For the global network training, the rotation factor and translation factor are set to $\pm 18^{\circ}$ and $\pm 3$ pixels, while settings of the local network training are the same as those of 300W local network training. As reported in Table IV, the proposed PicassoNet achieves the second best results in terms of NME. In respect of FR and AUC, the PicassoNet is comparable to Wing-ResNet50 [15] who takes about 4.12 GFLOPs, which is nearly $38\times$ times larger than our PicasssoNet. (Here, we neglect the FLOPs variation introduced by different output dimensions across datasets in the final fully connected layer.) Fig. 5 exhibits a couple of qualitative results on WFLW, which intuitively shows the ability of our model. A visualized accuracy–latency comparison is shown in Fig. 6, from which we can observe that the PicassoNet obtains a significantly higher efficiency. Nevertheless, it behaves poor in the pose subset. We speculate that this is because the cropped images of large poses lack enough discriminative information to locate landmarks. The flaw is strengthened in the cascaded framework where final localization depends on regional information and global constraints are not captured. We will analyze the phenomenon in Section IV-F.

*3) Evaluation on AFLW:* AFLW has 21 annotated landmarks, excluding contour keypoints. For a fair comparison, we align with the mainstream protocol that predicts 19 landmarks and discards ear points. Similar to the experiments on 300W, we conduct pose balancing besides common data augmentation. The augmentation settings for the global network training are [1.1, 1.16] for random scale, $\pm 20^{\circ}$ for random rotation, and $\pm 3$ pixels for random translation. Also, for the local network training, the corresponding settings are [0.9, 1.1], $\pm 6^{\circ}$, and $\pm 5$ pixels. Besides, we do not apply D-Loss because it is inaccurate to approximate tangent direction with AFLW's sparse landmark annotation. Table V lists the evaluation results. The PicassoNet yields the second best performance on AFLW-Full and the best performance on AFLW-Frontal.

### C. Analysis About Efficiency

To further verify the efficiency benefit of the proposed PicassoNet, we deploy the model on CPU (Intel Core i7-8700) to test the inference speed. For comparison, we also test popular architectures of existing methods, such as HRNet,[3] stacked hourglass,[4] and ResNet-50. According to Table VI, the proposed architecture has the least number of parameters and model size and runs much faster than the other networks, while the performance is competitive as analyzed in Section IV-B. To be more specific, PicassoNet runs $6.18\times$, $10.25\times$, and $19.96\times$ times faster than Wing-ResNet50, HRNet, and AWing on CPU. Besides, when implemented with mobile framework (TNN[5]), PicassoNet costs 7.1 ms/image on arm (Apple A10 Fusion at 2.37 GHz), namely, 140 frames/s, which is rapid enough to get involved in most of the mobile face applications.

### D. Verification About Grouping Search

To verify that the proposed grouping search indeed allocates computations for different groups, we analyze the initial architecture and optimized architecture of the local network. FLOPs assigned to different groups are reported in Table VII. In the proposed algorithm, the local network cuts down computations on nose and pay more attention to the other facial parts,

[3]https://github.com/HRNet/HRNet-Facial-Landmark-Detection
[4]https://github.com/protossw512/AdaptiveWingLoss
[5]https://github.com/Tencent/TNN

TABLE IV

EVALUATION RESULTS ON WFLW. WE USE BOLD AND UNDERLINE TO MARK THE BEST AND THE SECOND BEST SCORES

| Metric | Method | Year | FullSet | Pose | Expression | Illumination | Makeup | Occlusion | Blur |
|---|---|---|---|---|---|---|---|---|---|
| NME (%) | SDM [42] | 2013 | 10.29 | 24.10 | 11.45 | 9.32 | 9.38 | 13.03 | 11.28 |
| | CFSS [43] | 2015 | 9.07 | 21.36 | 10.09 | 8.30 | 8.74 | 11.76 | 9.96 |
| | DVLN [44] | 2017 | 6.08 | 11.54 | 6.78 | 5.73 | 5.98 | 7.33 | 6.88 |
| | LAB [7] | 2018 | 5.27 | 10.24 | 5.51 | 5.23 | 5.15 | 6.79 | 6.32 |
| | Wing [15] | 2018 | 5.11 | 8.75 | 5.36 | 4.93 | 5.41 | 6.37 | 5.81 |
| | AWing [26] | 2019 | **4.36** | **7.38** | **4.58** | **4.32** | **4.27** | **5.19** | **4.96** |
| | 3FabRec [45] | 2020 | 5.62 | 10.23 | 6.09 | 5.55 | 5.68 | 6.92 | 6.38 |
| | **PicassoNet(Ours)** | - | 4.82 | 8.61 | 5.14 | 4.73 | 4.68 | 5.91 | 5.56 |
| $FR_{0.1}$ (%) | SDM [42] | 2013 | 29.40 | 84.36 | 33.44 | 26.22 | 27.67 | 41.85 | 35.32 |
| | CFSS [43] | 2015 | 20.56 | 66.26 | 23.25 | 17.34 | 21.84 | 32.88 | 23.67 |
| | DVLN [44] | 2017 | 10.84 | 46.93 | 11.15 | 7.31 | 11.65 | 16.30 | 13.70 |
| | LAB [7] | 2018 | 7.56 | 28.83 | 6.37 | 6.73 | 7.77 | 13.72 | 10.74 |
| | Wing [15] | 2018 | 6.00 | 22.70 | 4.78 | 4.30 | 7.77 | 12.50 | 7.76 |
| | AWing [26] | 2019 | **2.84** | **13.50** | **2.23** | **2.58** | **2.91** | **5.98** | **3.75** |
| | 3FabRec [45] | 2020 | 8.28 | 34.35 | 8.28 | 6.73 | 10.19 | 15.08 | 9.44 |
| | **PicassoNet(Ours)** | - | 5.64 | 25.46 | 5.10 | 4.30 | 5.34 | 10.59 | 7.12 |
| $AUC_{0.1}$ | SDM [42] | 2013 | 0.300 | 0.023 | 0.229 | 0.324 | 0.312 | 0.206 | 0.239 |
| | CFSS [43] | 2015 | 0.366 | 0.063 | 0.316 | 0.385 | 0.369 | 0.269 | 0.304 |
| | DVLN [44] | 2017 | 0.455 | 0.147 | 0.389 | 0.474 | 0.449 | 0.379 | 0.397 |
| | LAB [7] | 2018 | 0.532 | 0.235 | 0.495 | 0.543 | 0.539 | 0.449 | 0.463 |
| | Wing [15] | 2018 | 0.550 | 0.310 | 0.496 | 0.541 | 0.558 | 0.489 | 0.492 |
| | AWing [26] | 2018 | **0.572** | **0.312** | **0.515** | **0.578** | **0.572** | **0.502** | **0.512** |
| | 3FabRec [45] | 2020 | 0.484 | 0.192 | 0.448 | 0.496 | 0.473 | 0.398 | 0.434 |
| | **PicassoNet(Ours)** | - | 0.554 | 0.255 | 0.510 | 0.554 | 0.556 | 0.460 | 0.486 |

TABLE V

NME (%) ON THE AFLW DATASET. "-" MEANS THAT THE CORRESPONDING FIGURE IS NOT REPORTED

| Method | LBF [46] | CFSS [43] | SAN [28] | LAB [7] | Wing [15] | ODN [21] | AWing [26] | 3FabRec [45] | **PicassoNet(Ours)** |
|---|---|---|---|---|---|---|---|---|---|
| AFLW-Full | 4.25 | 3.92 | 1.91 | 1.85 | 1.65 | 1.63 | **1.53** | 1.87 | 1.59 |
| AFLW-Frontal | 2.74 | 2.68 | 1.85 | 1.62 | - | 1.38 | 1.38 | 1.59 | **1.30** |

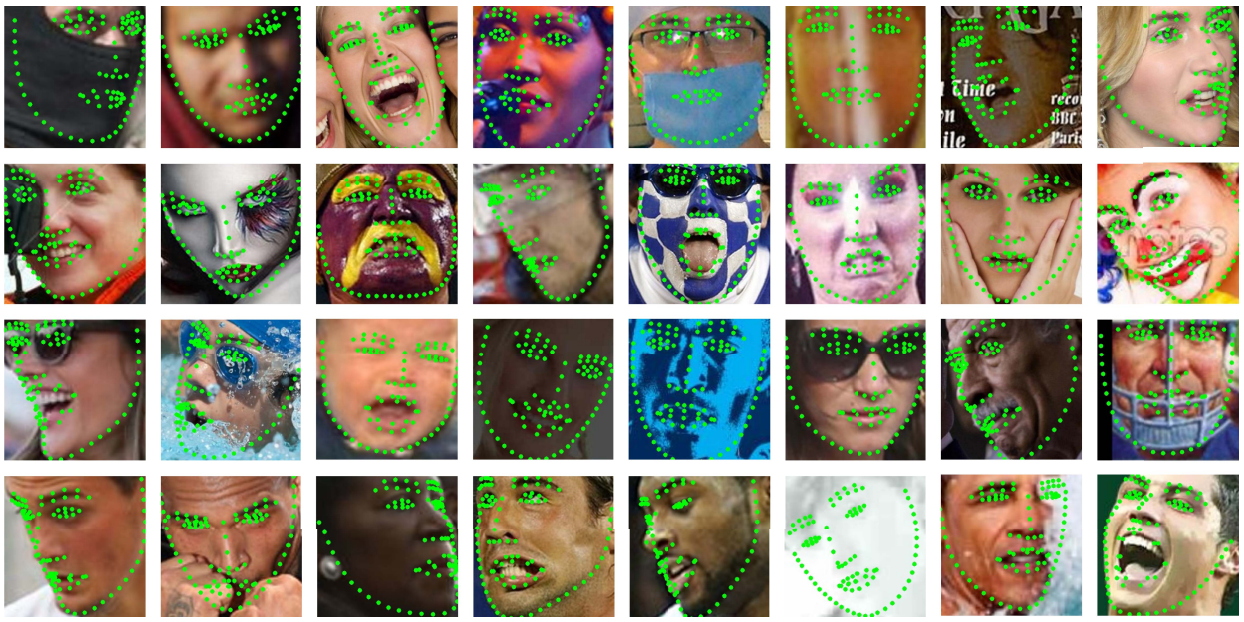


Fig. 5. WFLW result visualization. We can observe that most results produced by proposed model are satisfactory, which validates the effectiveness of our method.

which confirms our analysis above. We further visualize each layer's grouping state in Fig. 7, where the grouping state of FLGC keeps silent, while the proposed algorithm is capable of finding another architecture. The discoveries confirm our concern about softmax-based grouping search and demonstrate the effectiveness of the proposed algorithm.

TABLE VI

COMPARISON ABOUT MODEL COMPLEXITY AND RUN-TIME EFFICIENCY. FOR AWING AND HRNET, WE USE THE OFFICIAL IMPLEMENTATIONS TO OBTAIN THE STATISTICS. WE RUN EACH ALGORITHM FOR 200 TIMES TO AVOID JITTERING

| Method | Backbone | # Parameter (M) | Model size (MB) | FLOPs (G) | Latency (ms) |
|---|---|---|---|---|---|
| $\text{Wing}_{18}$ | ResNet-50 | 25.6 | 98.0 | 4.12 | 149 |
| $\text{AWing}_{19}$ | Stacked HG | 24.1 | 184 | 26.6 | 481 |
| $\text{HRNet}_{20}$ | HRNet-W18 | 9.60 | 37.3 | 4.71 | 247 |
| **PicassoNet(Ours)** | - | **1.96** | **7.75** | **0.106** | **24.1** |

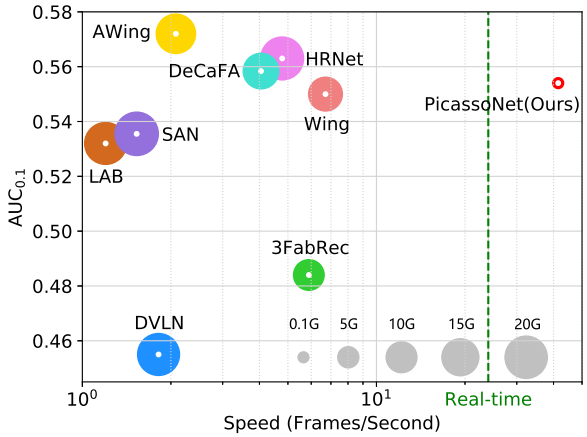



Fig. 6. AUC score and inference speed on the WFLW dataset (tested on i7-8700 at 3.20 GHz). The area of each circle notes FLOPs of the algorithm. The proposed PicassoNet yields comparable performance with Wing [15] and DeCaFA [10] and meanwhile runs 6.18× and 8.68× times faster.

TABLE VII

FLOPS FOR EACH GROUP IN THE LOCAL NETWORK. G1, G2, G3, AND G4 DENOTE THE FLOPS ASSIGNED TO LEFT EYE, RIGHT EYE, NOSE, AND MOUTH AREA, RESPECTIVELY. THE LARGE FLOPS OF G4 INITIALIZATION COME FROM WHOSE LINEAR LAYER PREDICTS THE MOST POINTS

| Grouping State | G1 (M) | G2 (M) | G3 (M) | G4 (M) |
|---|---|---|---|---|
| Gaussian initialization | 2.20 | 2.13 | 2.21 | 2.60 |
| FLGC | 2.20 | 2.13 | 2.21 | 2.60 |
| Ours | 2.27 | 2.21 | 1.98 | 2.68 |

TABLE VIII

FLOPS FOR EACH GROUP OF THE LOCAL NETWORK ACROSS DATASET. G1, G2, G3, AND G4 DENOTE THE FLOPS ASSIGNED TO LEFT EYE, RIGHT EYE, NOSE, AND MOUTH AREA, RESPECTIVELY

| Dataset | G1 (M) | G2 (M) | G3 (M) | G4 (M) |
|---|---|---|---|---|
| 300W | 2.27 | 2.21 | 1.98 | 2.68 |
| WFLW | 2.23 | 2.20 | 1.96 | 2.91 |
| AFLW | 2.10 | 2.03 | 1.83 | 3.02 |

In addition, we count each group's FLOPs across dataset to verify the consistency about the searched results. From Table VIII, we can observe that the searched results are roughly consistent for all the datasets. The FLOPs of regions subject to that "mouth > eye+ > nose," which matches what the toy experiment indicated in Table II.

To verify the searched result is optimal, we further compare the proposed grouping search algorithm with manually divided groups and random divided groups. The results are reported in Table IX, where the searched grouping yields the best results and naturally avoids tedious handcrafts.

TABLE IX

COMPARISON AGAINST MANUAL SET AND REDOM SET ON THE 300W DATASET. MANUAL SETS A AND B ARE DIFFERENT FLOPS SET FOLLOWING THE RULES INDICATED IN TABLE VIII

| Method | Common | Challenge | Fullset |
|---|---|---|---|
| **Searched** | **2.41** | **4.94** | **2.91** |
| Manual set A | 2.52 | 4.98 | 3.00 |
| Manual set B | 2.50 | 4.97 | 2.98 |
| Random | 2.59 | 5.08 | 3.08 |

TABLE X

LOCAL NETWORK'S NME (%) ON 300W. RI DENOTES WHETHER THE EXPERIMENT ADOPTS RECOMBINED INPUTS. ALSO, G IS GROUPING MANNER, AND "×", "EVEN," AND "ADAPTIVE" REPRESENT NO GROUPING, EVEN GROUPING, AND ADAPTIVE GROUPING, RESPECTIVELY

| RI | G | Common | Challenge | Fullset |
|---|---|---|---|---|
| × | - | 2.47 (-) | 5.16 (-) | 3.00 (-) |
| ✓ | × | 2.57 (+4.0%) | 5.33 (+3.3%) | 3.11 (+3.7%) |
| ✓ | even | 2.50 (+1.2%) | 5.05 (-2.1%) | 3.00 (-) |
| ✓ | adaptive | **2.41 (-2.4%)** | **4.94 (-4.3%)** | **2.91 (-3.0%)** |

TABLE XI

EVALUATION OF OPTIMIZING DIRECTIONS ON 300W. WE TRAIN THE GLOBAL NETWORK VIA OPTIMIZING WING LOSS AND D-LOSS. $\text{NME}_t$ AND $\text{NME}_n$ DENOTE THE NME ON TANGENTIAL DIRECTION AND NORMAL DIRECTION, RESPECTIVELY

| Loss function | $\alpha$ | NME (%) | $\text{NME}_t$ | $\text{NME}_n$ |
|---|---|---|---|---|
| Wing | - | 4.19 | 3.04 | 1.62 |
| **D-Loss** | 0.6 | **3.99** | **2.95** | **1.49** |
| D-Loss | 1.0 | 4.20 | 3.05 | 1.62 |
| D-Loss | 1.5 | 4.23 | 3.01 | 1.67 |

### *E. Ablation Study*

We conduct ablation experiments on 300W about employing a generalized local network and utilizing the grouping search method. The experiments and scores are listed in Table X. For experiment without reconstructed inputs or adaptive grouping search, we train four subnetworks for local stage, which in fact is the conventional cascaded algorithm. In particular, each subnetwork takes a quarter as much local network computations to maintain equal computation complexity. The results show that combining reconstructed inputs and adaptive

TABLE XII
NME ALONG ANGLE VARIATION ON WFLW. THE ANGLE IS COMPUTED BY SUMMING OVER PITCH AND YAW

| Angle (°) | 0∼10 | 10∼20 | 20∼30 | 30∼40 | 40∼50 | 50∼60 | 60∼70 | 70∼80 | 80∼90 | >90 |
|---|---|---|---|---|---|---|---|---|---|---|
| NME | 3.47 | 3.63 | 3.93 | 4.8 | 5.93 | 7.41 | 7.49 | 8.67 | 11.31 | 10.57 |

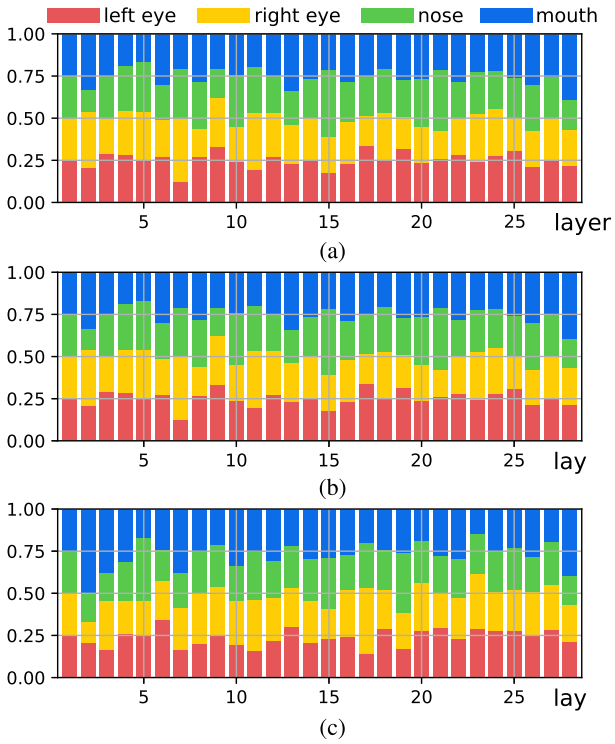



Fig. 7. Grouping states of FLGC and the proposed method. The bars in four colors denote the channel proportion of individual facial parts in each layer. From bottom to top, left eye, right eye, nose, and mouth. (a) Initial grouping state with Gaussian initialization. (b) Grouping state found by FLGC. (c) Grouping state found by the proposed algorithm.

grouping search obtains the best performance, which verifies our assumption that submodules of individual facial parts might differ in requisite computation complexity. Nevertheless, it is ineffectual to adopt reconstructed input only without grouping. This is because all the spatial isolated facial parts are mixed to generate the final output, and the interactions are misleading to each local regression task. We can also notice that even grouping of single local network and multiple independent subnetworks shares a similar performance, which supports our generalization that models submodules using convolution groups.

Furthermore, we report the quantitative verification about the proposed D-Loss in Table XI, where the global network optimizing D-Loss ($\alpha = 0.6$) obtains the best performance on NME. When $\alpha = 1$, D-Loss behaves the same as Wing loss, which agrees with the degeneration of D-Loss discussed in Section III-D. From Table XI, we can also observe that the normal NME declines when the model penalizes more on the normal direction.

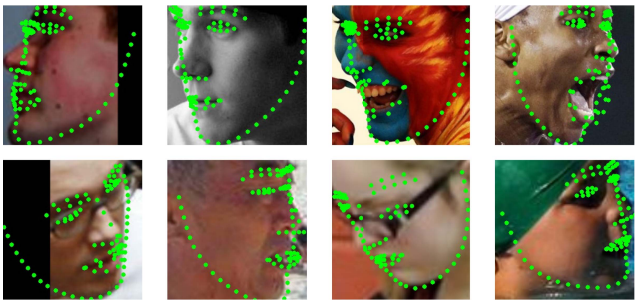


Fig. 8. Failure cases in WFLW experiments, these weak results reveal that there are still many challenges standing in the zkeypoint detection field.

### F. Failure Analysis

We visualize the landmarks predicted by the proposed method, most of which are satisfying, as shown in Fig. 5. While there also exist bad cases, most of them are in extreme poses, as reported in Fig. 8, which as well can be confirmed numerically from WFLW experiments. Table XII shows accuracy at different pose angles, where NME increases rapidly when the Euler angle is larger than 30°. We speculate that the phenomenon comes from the global network's weakness about pose variation and less information in the local area. From this point of view, utilizing global topological relationship might help to tackle pose issues.

## V. CONCLUSION

In this work, we present the PicassoNet, a facial landmark localization algorithm with decent accuracy and high runtime efficiency. Adopting global–local cascaded network as the baseline, we generalize regional submodules using convolution groups and integrate them into a single compact network. The local network receives a single input tensor reconstructed by different facial parts and decomposes the compound input during forward inference. A novel grouping search algorithm is proposed to optimize the structure of the local network, which adaptively allocates computations for individual facial parts. In addition, we propose D-Loss to solve the boundary fit problem of the predicted landmarks. Experiments on three popular datasets demonstrate the high efficiency and satisfactory performance of the proposed methods.

## REFERENCES

[1] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4690–4699.
[2] X. Wang, S. Zhang, S. Wang, T. Fu, H. Shi, and T. Mei, "Mis-classified vector guided softmax loss for face recognition," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 12241–12248.

[3] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2Face: Real-time face capture and reenactment of RGB videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2387–2395.

[4] W. Chu, Y. Tai, C. Wang, J. Li, F. Huang, and R. Ji, "SSCGAN: Facial attribute editing via style skip connections," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 414–429.

[5] P. Dou, S. K. Shah, and I. A. Kakadiaris, "End-to-end 3D face reconstruction with deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5908–5917.

[6] L. Tran and X. Liu, "Nonlinear 3D face morphable model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7346–7355.

[7] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou, "Look at boundary: A boundary-aware face alignment algorithm," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2129–2138.

[8] A. Kumar *et al.*, "LUVLi face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8236–8246.

[9] M. Kowalski, J. Naruniec, and T. Trzcinski, "Deep alignment network: A convolutional neural network for robust face alignment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 88–97.

[10] A. Dapogny, M. Cord, and K. Bailly, "DeCaFA: Deep convolutional cascade for face alignment in the wild," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6893–6901.

[11] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 1–16.

[12] 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)*, Jul. 2017, pp. 3317–3326.

[13] P. Chandran, D. Bradley, M. Gross, and T. Beeler, "Attention-driven cropping for very high resolution facial landmark detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5861–5870.

[14] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[15] Z.-H. Feng, J. Kittler, M. Awais, P. Huber, and X.-J. Wu, "Wing loss for robust facial landmark localisation with convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2235–2245.

[16] 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)*, Nov. 2011, pp. 2144–2151.

[17] 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*, Dec. 2013, pp. 397–403.

[18] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks)," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1021–1030.

[19] J. Yang, Q. Liu, and K. Zhang, "Stacked hourglass network for robust facial landmark localisation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 79–87.

[20] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 483–499.

[21] M. Zhu, D. Shi, M. Zheng, and M. Sadiq, "Robust facial landmark detection via occlusion-adaptive deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3486–3496.

[22] J. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021.

[23] D. Merget, M. Rock, and G. Rigoll, "Robust facial landmark detection via a fully-convolutional local-global context network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 781–790.

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

[25] 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*, Dec. 2013, pp. 386–391.

[26] X. Wang, L. Bo, and L. Fuxin, "Adaptive wing loss for robust face alignment via heatmap regression," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6971–6981.

[27] S. Lai, Z. Chai, S. Li, H. Meng, M. Yang, and X. Wei, "Enhanced normalized mean error loss for robust facial landmark detection," in *Proc. BMVC*, 2019, p. 111.

[28] 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.*, Jun. 2018, pp. 379–388.

[29] X. Dong and Y. Yang, "Teacher supervises students how to learn from partially labeled images for facial landmark detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 783–792.

[30] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[32] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.

[33] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet v2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.

[34] X. Wang, M. Kan, S. Shan, and X. Chen, "Fully learnable group convolution for acceleration of deep neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9049–9058.

[35] Z. Zhang *et al.*, "Differentiable learning-to-group channels via groupable convolutional neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3542–3551.

[36] P. Ren *et al.*, "A comprehensive survey of neural architecture search: Challenges and solutions," 2020, *arXiv:2006.02903*.

[37] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," 2016, *arXiv:1611.01144*.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[39] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A face detection benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5525–5533.

[40] P. Adam *et al.*, "Automatic differentiation in PyTorch," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 1–4.

[41] H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable architecture search," in *Proc. 7th Int. Conf. Learn. Represent. (ICLR)*, New Orleans, LA, USA, May 2019, pp. 1–13. [Online]. Available: https://openreview.net/forum?id=S1eYHoC5FX

[42] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 532–539.

[43] 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)*, Jun. 2015, pp. 4998–5006.

[44] 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)*, Jul. 2017, pp. 150–159.

[45] B. Browatzki and C. Wallraven, "3FabRec: Fast few-shot face alignment by reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6110–6120.

[46] 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.*, Jun. 2014, pp. 1685–1692.

[47] 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)*, Jun. 2016, pp. 4177–4187.

[48] L. Yue, X. Miao, P. Wang, B. Zhang, X. Zhen, and X. Cao, "Attentional alignment networks," in *Proc. BMVC*, 2018, vol. 2, no. 6, p. 7.

[49] X. Dong, S.-I. Yu, X. Weng, S.-E. Wei, Y. Yang, and Y. Sheikh, "Supervision-by-registration: An unsupervised approach to improve the precision of facial landmark detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 360–368.

[50] X. Miao, X. Zhen, X. Liu, C. Deng, V. Athitsos, and H. Huang, "Direct shape regression networks for end-to-end face alignment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5040–5049.

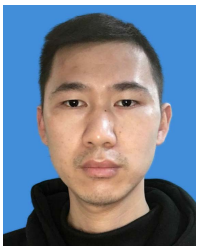
[51] Z. Tang, X. Peng, S. Geng, Y. Zhu, and D. N. Metaxas, "CU-Net: Coupled U-Nets," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Newcastle, U.K., Sep. 2018, p. 305. [Online]. Available: http://bmvc2018.org/contents/papers/0338.pdf

[52] J. Robinson, Y. Li, N. Zhang, Y. Fu, and S. Tulyakov, "Laplace landmark localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10103–10112.

[53] R. Valle, J. M. Buenaposada, A. Valdés, and L. Baumela, "Face alignment using a 3D deeply-initialized ensemble of regression trees," *Comput. Vis. Image Understand.*, vol. 189, Dec. 2019, Art. no. 102846.

**Tiancheng Wen** received the B.S. degree from Xi'an Jiaotong University, Xi'an, China, in 2018, where he is currently pursuing the M.S. degree with the Smiles Laboratory.

His research interests are knowledge distillation, face recognition, and face alignment.



**Zhonggan Ding** received the B.S. degree from the Wuhan University of Technology, Wuhan, China, in 2013, and the M.S. degree from the Huazhong University of Science and Technology, Wuhan, in 2016.

His research interests focus on computer vision and artificial intelligence, specifically on the topic of scene classification, face alignment, object detection, and makeup transfer.

**Yongqiang Yao** received the M.S. degree from Beihang University, Beijing, China, in 2018.

His research interests are face alignment and human pose estimation.

**Yaxiong Wang** received the B.S. degree from Lanzhou University, Lanzhou, China, in 2015, and the Ph.D. degree from the School of Software Engineering, Xi'an Jiaotong University, Xi'an, China, in 2021.

His current research interests include few-shot learning, cross-media retrieval, generative learning, and superpixel segmentation.

**Xueming Qian** (Member, IEEE) received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the Ph.D. degree from the School of Electronics and Information Engineering, Xi'an Jiaotong University, in 2008.

He was an Assistant Professor with Xi'an Jiaotong University, where he was an Associate Professor from 2011 to 2014, and is currently a Full Professor. He was a Visiting Scholar with Microsoft Research Asia, Beijing, China, from 2010 to 2011. He is also the Director of the SMILES Laboratory, Xi'an Jiaotong University. His research interests include social media big data mining and search.

Dr. Qian received the Microsoft Fellowship in 2006 and the Outstanding Doctoral Dissertations of Xi'an Jiaotong University and Shaanxi Province in 2010 and 2011, respectively.