Efficient Face Alignment with Fast Normalization and Contour Fitting Loss

ZHIWEI LIU, XIANGYU ZHU, MING TANG, ZHEN LEI, and JINQIAO WANG, University of Chinese Academy of Sciences, National Lab of Pattern Recognition, Institute of Automation, CAS, Beijing

Face alignment is a key component of numerous face analysis tasks. In recent years, most existing methods have focused on designing high-performance face alignment systems and paid less attention to efficiency. However more face alignment systems are now applied on low-cost devices, such as mobile phones. In this article, we design a common efficient framework that can team with any face alignment regression network and improve the overall performance with nearly no extra computational cost. First, we discover that the maximum regression error exists in the face contour, where landmarks do not have distinct semantic positions, and thus are randomly labeled along the face contours in training data. To address this problem, we propose a novel contour fitting loss that dynamically adjusts the regression target during training so the network can learn more accurate semantic meanings of the contour landmarks and achieve better localization performance. Second, we decouple the complex sample variations in face alignment task and propose a Fast Normalization Module (FNM) to efficiently normalize considerable variations that can be described by geometric transformation. Finally, a new lightweight network architecture named Lightweight Alignment Module (LAM) is also proposed to achieve fast and precise face alignment on mobile devices. Our method achieves competitive performance with state-of-the-arts on 300W and AFLW2000-3D benchmarks. Meanwhile, the speed of our framework is significantly faster than other CNN-based approaches.

CCS Concepts: • **Computing methodologies** → **Artificial intelligence**; • **Artificial intelligence** → *Computer vision*; • **Computer vision** → Computer vision tasks; • **Computer vision tasks** → Biometrics;

Additional Key Words and Phrases: Face alignment, convolutional neural networks, real-time, semantic meaning

ACM Reference format:
Zhiwei Liu, Xiangyu Zhu, Ming Tang, Zhen Lei, and Jinqiao Wang. 2019. Efficient Face Alignment with Fast Normalization and Contour Fitting Loss. *ACM Trans. Multimedia Comput. Commun. Appl.* 15, 3s, Article 89 (October 2019), 16 pages.
https://doi.org/10.1145/3338842

1 INTRODUCTION

Face alignment algorithms try to detect the facial landmarks by using the facial appearance in 2D face images. It plays a fundamental role in many face analysis tasks, such as face recognition, expression estimation, and 3D face reconstruction. Recently, face alignment methods have been

This work is supported by the Natural Science Foundation of China (Grant nos. 61772527, 61806200, 61876178, 61806196).
Authors' addresses: Z. Liu, X. Zhu, M. Tang, Z. Lei, and J. Wang (corresponding author), University of Chinese Academy of Sciences, National Lab of Pattern Recognition, Institute of Automation, CAS, Beijing, 95 Zhongguancun East Road, Beijing, 100190, China; emails: {zhiwei.liu, xiangyu.zhu, tangm, zlei, jqwang}@nlpr.ia.ac.cn.

1551-6857/2019/10-ART89 $15.00
https://doi.org/10.1145/3338842

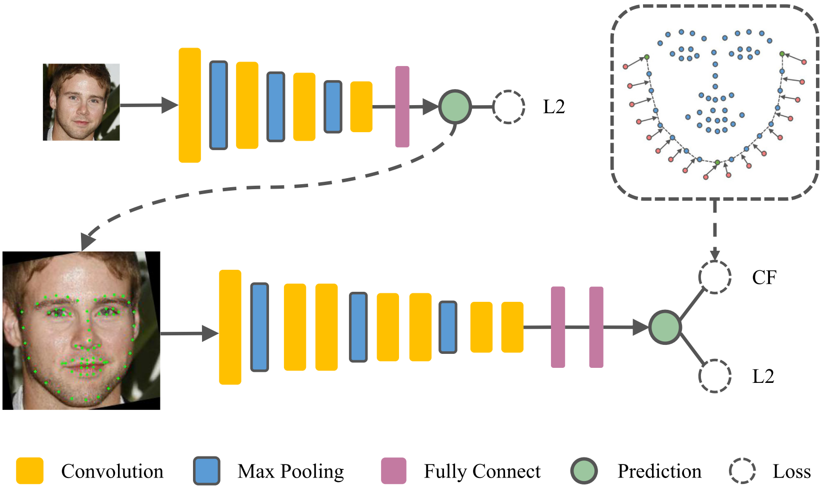


Fig. 1. Overview of our framework. The upper network is a Fast Normalization Module that can efficiently normalize translations, scalings, and in-plane rotations. Then the normalized face image is input to our main network to predict the final landmarks. We use the lightweight CNN at both stages and apply CF loss to train our main network.

required to handle the complicated pose, illumination, and expression variations in unconstrained environment and to run on low-cost devices such as mobile at real-time speed.

Thanks to the powerful fitting ability of Convolutional Neural Networks (CNNs), CNNs have been successfully applied to face alignment task, regarding the landmark localization task as a regression problem. Deep learning–based methods regress landmark coordinates [25, 35, 43, 45] or heatmaps [10, 40] from facial images. However, they relied on a heavy network [10, 12, 25, 37, 40] to get powerful fitting ability with unsatisfactory efficiency. Few high-performance methods can achieve real-time speed on CPU, certainly not on low-cost handheld devices such as mobile phones. Therefore, designing a CNN-based framework with satisfactory performance and high efficiency is crucial for face alignment applications. To achieve this goal, we propose to improve existing face alignment framework in two aspects: designing more powerful loss function without bringing computational cost and designing a efficient framework that allocates the limited computing resource reasonably.

In this article, we first investigate the localization errors of existing face alignment methods and indicate that most detection errors come from the contour landmarks. We further compare the predicted landmarks with the ground-truth ones and notice that the positions of labeled landmarks have some potential randomness. And the randomness depends on whether the landmarks possess distinct semantic positions or not. Some landmarks have distinct semantic positions that usually distribute on the corners of face structure, such as eye corner, mouth corner. For this kind of landmark, annotators are able to accurately identify their locations by their texture appearance. In contrast, some landmarks (e.g., those evenly distributed along the face contour) do not have clear and accurate definition. Therefore, it can make human annotators confused about the positions of weak semantic points, and it is inevitable for annotators to introduce random noises during annotating. The inconsistent and imprecise annotations can mislead CNN training and cause degraded performance. Since the face contour curve is much longer than other curves, the random noise is most serious in contour landmarks. To resolve this problem, we propose a novel contour fitting regression loss (CF Loss). The contour fitting loss automatically adjusts the regression target during training and helps the network better understand the semantic position of contour landmarks. It is beneficial for searching for the better local optima and paying more attention to localize the points that possess strong semantic meaning. Experiments show that our novel contour fitting loss is effective for improving performance of face alignment.

To better allocate the limited computing resource, we systematically analyze the effects of different pose variations and present an efficient two-stage face alignment framework with a Fast Normalization Module (FNM) followed by an unprecedented lightweight face alignment network named Lightweight Alignment Module (LAM). Different from the common landmark localization methods that adopt a heavy network to deal with all the variations [18, 23, 38, 40], we demonstrate that under strict computation limit, it is more effective and efficient to divide the face alignment into two subtasks: image normalization and landmark localization. First, FNM coarsely localizes the landmarks with tiny cost, normalizing the scaling, translation, and in-plane rotation of the input images. Then the LAM trained by our novel contour fitting loss accurately localizes the landmarks. Although two-stage framework has been applied in previous works [14, 25], they did not consider the efficiency of the normalization stage and adopted heavy networks to normalize the image, generating a lot of redundant computation. In this article, we demonstrate that a simple and tiny network is able to finish the normalization task, and this normalization is especially important when the computation resource is limited.

This work aims to provide a surprisingly efficient baseline method to alleviate the impact of simply described variations and weak semantic points because LAM can be replaced by any CNN regression network. Our method is a common efficient face alignment framework that can significantly improve the overall performance with nearly no extra computational cost. We believe that our framework is helpful for inspiring and evaluating new ideas for the face alignment field. Extensive experiments on 300W and AFLW2000-3D datasets demonstrate the effectiveness of our method and achieve competitive performance with other state-of-the-art methods.

The main contributions of our work include:

- According to the different semantic positions, we propose a novel contour fitting regression loss that helps the network better understand the semantic positions of contour landmarks to improve the localization performance.
- We decouple the complex sample variations in face alignment task and design a common two-stage alignment framework with an FNM and an LAM to localize landmarks accurately and efficiently. The two-stage architecture is found very effective when the computation resource is limited.
- Our method achieves competitive performance with the state-of-the-art on 300W and AFLW2000-3D benchmarks and the speed is also significantly faster than other CNN-based face alignment approaches.

2 RELATED WORK

Face alignment has achieved remarkable successes during the past decades, which can be divided in several ways. Traditional approaches are based on statistical models such as Active shape models (ASM) [7] and Active appearance models (AAM) [8], which perform well in some constrained scenarios. Recently, regression-based methods, especially cascaded regression methods, signicantly boost the performance of face alignment. The most efficient cascaded regression methods need an initial shape and regress the current offset via boosting of weak regressors [6, 11, 39] or the shadow models [30, 42]. The representative approaches are Supervised Descent Method (SDM) [39] and Coarse-to-Fine Auto-Encoder Network (CFAN) [42], which cascades several stacked auto-encoders in a coarse-to-fine way to refine the shape stage-by-stage. Although cascaded methods achieve promising performance on both controlled and wild settings with high speed, most of them are based on linear or shallow models that might be insufficient for modeling the complex nonlinearity problem, especially under the challenging wild scenario. Furthermore, extraction of the hand-crafted features like SIFT [24] for dense points also consumes lot of time. So it is not easy to extend cascaded regression method to tackle with challenge task in most practical applications.

In the past three years, convolutional neural networks (CNN) achieved very impressive results on many computer vision tasks, including face alignment. Sun et al. [35] propose to cascade several DCNN to predict the shape stage-by-stage. Zhang et al. [45] proposed a single CNN and jointly optimize facial landmark detection together with other tasks of facial attributes, further enhancing the speed and performance. The methods above use shadow CNN models to directly regress facial landmarks, which is difficult to cope with the complex task with dense landmarks and large pose variations. Many popular semantic segmentation and human pose estimation frameworks can also be used for face alignment [10, 40]. They predict a heatmap which contains existing probability of the corresponding landmark. These methods usually need lots of model parameters and much computation load due to the up-sampling and deconvolution operation. In this article, we focus on designing an efficient framework with high accuracy. Therefore, we use CNN to directly regress the landmark coordinates like traditional CNN-based approaches. Feng et al. [14] study different regression loss and design a piece-wise loss function that paid attention to the small- and medium-range errors and alleviate impact of outliers. Different from this method that develops loss function from the theoretical perspective, we analyze different semantic meanings of various landmarks in the human face and proposed a novel contour fitting loss. Our contour fitting loss pays more attention to localize the points that possess a strong semantic meaning and help CNN search for the better local optima.

Two recent works, LAB [37] and SBR [12], are related to the problem caused by weak semantic points. LAB trains a facial boundary heatmap estimator and incorporates it into the main landmark regression network to provide the facial geometric structure, leading to improved performance. SBR proposes a registration loss that uses the coherency of optical flow from adjacent frames as its supervision. They all utilize extra information to help CNN better understand the face structure. However, compared with our method, they have many disadvantages: First, the regression networks of LAB and SBR still use ground-truth as their optimization target. Therefore, LAB and SBR do not intrinsically address the problem caused by weak semantic points, because the degraded accuracy is actually derived from the random noise in annotated contour points. Second, LAB and SBR both apply complex network (e.g., hourglass) to regress heatmaps in their framework, which are computationally expensive and cannot be applied on platforms with limited computational resources. Third, the optical flow used in SBR is not always credible in unconstrained environments. In this work, we propose a common efficient face alignment framework that can team with any backbone and alleviate the problem caused by weak semantic points.

Recently, model compression and efficient network architecture design have become a hot topic. One of the representative approaches is weight pruning [17] and weight quantization [9, 29]. These approaches are effective, because deep networks often have a substantial number of redundant weights that can be pruned or quantized without sacrificing (and sometimes even improving) accuracy. Many recent studies have explored efficient architectures that can be trained end-to-end, such as MobileNet [16], ShuffleNet [44], and Neural Architecture Search (NAS) networks [49]. In this article, we follow the second way to develop the lightweight efficient face alignment framework that can be trained from scratch. We systematically analyze the characteristics of face alignment task and use many effective strategies to design our efficient framwork with the Fast Normalization Module (FNM).

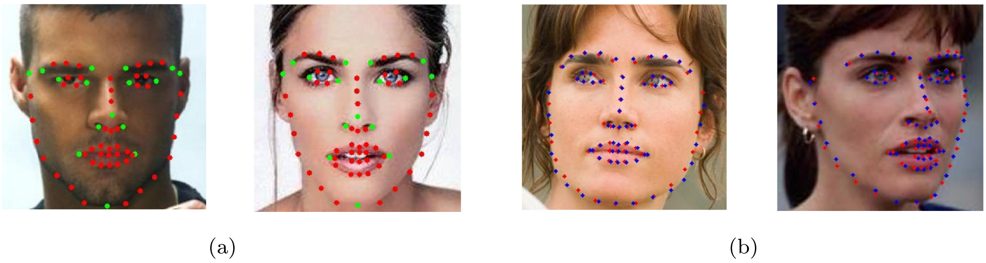


Fig. 2. (a) Strong semantic point and weak semantic point. Green points represent the strong semantic points that can be accurately localized by the annotator. Red points represents the weak semantic points, and their position can only be estimated. (b) Some cases where the contour landmarks have been good enough but there are still large point-to-point errors.

3 CONTOUR FITTING LOSS

For CNN-based face alignment methods trained with the common L2 loss, we observe the training loss curves of different face areas and find that the error of face contour points is significantly higher than that of other facial feature points. The main reason is that face contour points have no strict semantic positions. They are just defined to be evenly distributed along the face contour. Hence, the annotators cannot judge the exact location of these points. So there are potential random factors in the labelled face contour landmarks, during training, even though the predicted contour points have been accurately located on the contour curve, there are still large point-to-point errors due to the randomness of the regression targets. Considering the above problems of weak semantic points, existing methods apply Iterative Closest Point (ICP) algorithms [3] to the surface registration [1] and 3DMM fitting [31] task. ICP algorithms are employed to minimize the difference between two clouds of points by searching for the closest points to the template vertices on the target surface. Note that reducing the error between the output of CNN and ground-truth is similar with the process of point cloud registration. Inspired by this, we propose a novel contour fitting loss (CF loss) to help CNN better understand the semantic positions of face contour landmarks. Contour fitting loss automatically adjusts the regression target during training and successfully improves the overall localization accuracy.

In this section, we introduce a contour fitting loss to resolve the non-strict semantic positions of facial contour landmarks. We first analyze the semantic positions of each landmark and propose that landmarks should be divided into strong semantic points and weak semantic points. Strong semantic points indicate the points that have distinct semantic positions, such as nose tip, eye corner, and mouth corner. These points distribute on the apexes and corners of the face structure, so they can be exactly localized by annotators. Weak semantic points do not have distinct semantic positions; they are just required to evenly distribute along some curves of the face structure, such as eye line, lip line, and face contour. For these points, the annotators cannot localize their positions and just label them on the curve and coarsely make them evenly distributed. Since the face contour curve is much longer than other curves, we suppose the contour landmarks have the weakest semantic meanings and are extremely difficult to accurately label. As a result, there is severe randomness in the labelled positions of face contour landmarks.

To verify the above analysis, we analyze the offsets between the labelled points and the predicted points. As shown in Figure 2(b), despite the predicted face contour curve being pretty close to the real contour curve, the offsets between the predicted contour points and the ground-truth points also bring considerable training errors during training. The error brought by the contour points is larger than that brought by the facial feature points, and the optimization direction might be dominated by these contour points. Moreover, the error from the contour is difficult to be reduced due to the randomness of the regression target. This case could make the optimization reach the plateau. To alleviate the effect of the weak semantics of contour points and further improve the performance, we relax the point-to-point loss to point-to-curve loss and dynamically adjust the regression target for weak semantic points.

The target of CNN-based face alignment method is to find a nonlinear mapping:

$$\Phi : \mathcal{I} \rightarrow \mathbf{s}, \tag{1}$$

where $\mathbf{s} \in \mathbb{R}^{2N}$ represents the output shape vector predicted by the input image $\mathcal{I} \in \mathbb{R}^{H \times W \times 3}$. The shape vector is composed of $\mathbf{s} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N]^T$. $\mathbf{p}_i = (x_i, y_i)$ represents the coordinates vector of the point and N is the number of landmarks. Existing methods mainly adopt L2 loss to regress the landmark coordinates, which is defined as:

$$L_{L2} = \sum_{i=1}^{L} \|\mathbf{p}_i - \mathbf{p}_i^*\|_2^2, \tag{2}$$

where $\mathbf{p}$ represents the point predicted by CNN and $\mathbf{p}^*$ represents the ground-truth points if we separate the contour points from the facial feature points:

$$L_{L2} = \sum_{i=1}^{N_c} \left\|\mathbf{p}_{c_i} - \mathbf{p}_{c_i}^*\right\|_2^2 + \sum_{i=1}^{N_f} \left\|\mathbf{p}_{f_i} - \mathbf{p}_{f_i}^*\right\|_2^2, \tag{3}$$

where N_c and N_f represent the number of contour points $\mathbf{p}_c$ and other facial feature points $\mathbf{p}_f$. Note that L2 loss gives equal attention to all the landmark points. When the predicted contour points are located on the contour curve but are far from its ground-truth positions, their optimization directions will be along the contour curve and produce larger errors than other facial feature points, as shown in Figure 2(b). At this time, the overall optimization direction might be dominated by these insignificant errors from contour points and stop optimizing the facial feature points.

To address this problem, we reshape the common L2 loss to a novel contour fitting loss, which optimizes point-to-curve distances rather than point-to-point distances. Specifically, we first get the real face contour curve by connecting all adjacent contour points. For each predicted face contour point, we connect its ground-truth points and two adjacent points to determine a target curve, as shown on the right side of Figure 3. During training, the contour fitting loss finds the nearest point on the target curve for each predicted contour point and regards them as the new regression target as depicted on the left side of Figure 3, which can be defined as:

$$\begin{aligned} L_{CF} &= \sum_{i=1}^{N_c} \|\mathbf{p}_{c_i} - \mathbf{t}_i^*\|_2^2 + \sum_{i=1}^{N_f} \left\|\mathbf{p}_{f_i} - \mathbf{p}_{f_i}^*\right\|_2^2 \\ \textit{where} \qquad \mathbf{t}_i^* &= \underset{\mathbf{t}_{ij} \in \mathcal{N}(\mathbf{p}_{c_i})}{\arg\min}\ d(\mathbf{p}_{c_i}, \mathbf{t}_{ij}), \end{aligned} \tag{4}$$

where $\mathcal{N}(\mathbf{p}_{c_i})$ represents the point set of the target curve. $\mathbf{t}_{ij}$ represents jth points in the point set. The new target point $\mathbf{t}_i^*$ has the minimum Euclidean distance to the predicted point. In our implementation, we adopt the Bresenham algorithm to get the point set $\mathcal{N}(\mathbf{p}_{c_i})$ of the target curve. Besides, we combine the contour fitting loss with L2 loss by adding a hyper-parameter λ. The final loss function is written as:

$$L = \lambda L_{CF} + (1 - \lambda) L_{L2}. \tag{5}$$

The new loss is also able to regress the contour points to fit the contour curve and make a good balance of the contour points and facial feature points. Note that the target curves only depend on the ground-truth points; we calculate them by Bresenham offline before training and just load it

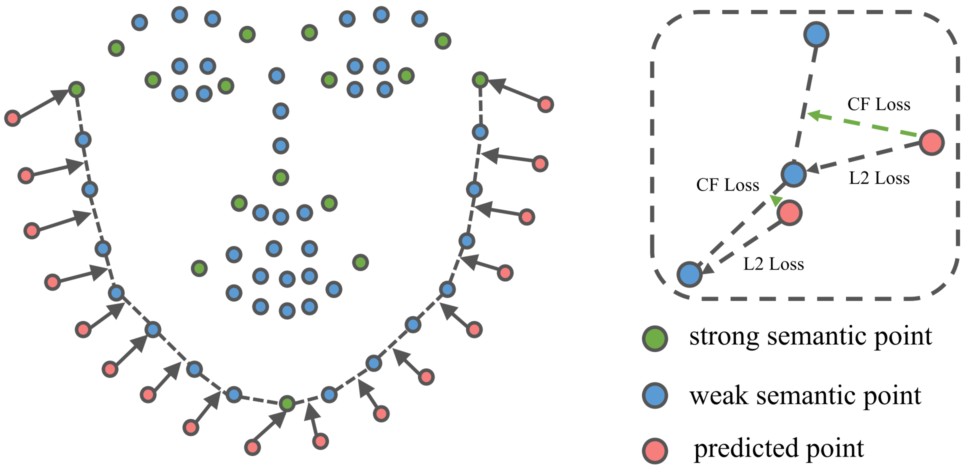
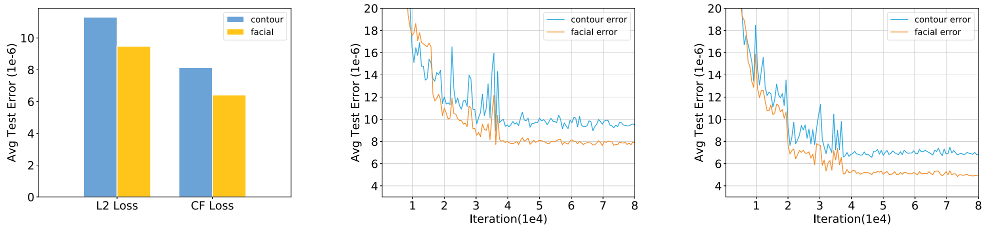


Fig. 3. Illustration of contour fitting loss. The left side shows the optimization direction of the contour points. Right side shows the detail, which indicates the target curve and the difference between CF Loss and L2 Loss.



(a) (b) (c)

Fig. 4. (a) shows the error of contour points and facial feature points by L2 loss and CF loss, respectively. (b) shows the test error curve along the training process by the L2 loss. (c) shows the test error curve along the training process by the CF loss. Note that the CF loss reduces the error of both contour landmarks and facial feature landmarks.

during optimization. Hence, the proposed contour fitting loss just slightly increases training time and does not increase inference time cost.

At the early stage of training, the predicted landmarks have large deviations with the real landmarks due to the under-fitting of CNN. Directly using contour fitting loss would lead CNN to output non-even contour points. As a result, we adopt the L2 loss at the beginning and replace L2 loss by contour fitting loss after 20K iterations. Figure 4(b) and Figure 4(c) show the testing error curve along the training process. We can observe that the contour point error is close to the facial feature points error at the early stage. While after 20K iterations when the training starts to converge by the L2 loss, the contour fitting loss continuously optimizes the CNN to a better local optima, getting more accurate facial features and contour landmarks. In general, the contour fitting loss provides a better optimizing direction and brings considerable performance promotion.

4 EFFICIENT ALIGNMENT FRAMEWORK

In this section, we introduce the network structure of our efficient two-stage face alignment framework. In the first stage, we normalize the scaling, translation, and in-plane rotation with a well-designed Fast Normalization Module (FNM). In the second stage, we further design a Lightweight

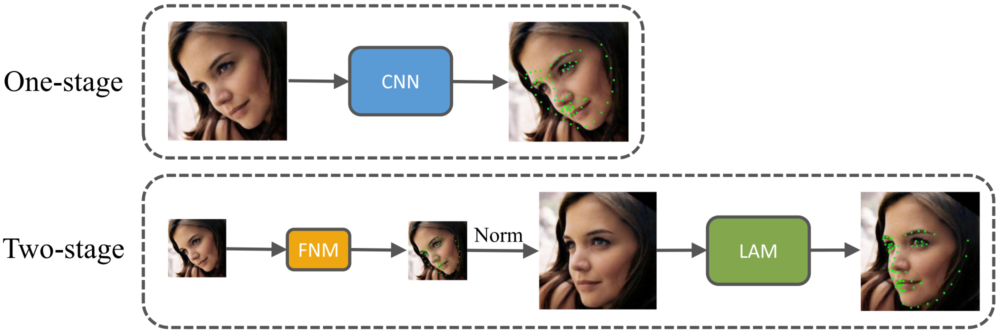


Fig. 5. Illustration of Fast Normalization Module (FNM). The upper framework is the traditional method that uses one powerful CNN to deal with all kinds of pose variations. This cannot work when we apply the lightweight CNN. The lower one represents our proposed efficient two-stage framework with FNM.

Alignment Module (LAM) to achieve fast and precise face alignment on mobile devices. Figure 1 shows the overview of our framework. Note that the contribution of FNM and LAM is not mainly based on any theoretical evidence. It is also based on simple techniques and validated by comprehensive ablation experiments.

4.1 Fast Normalization Module

In real applications, face alignment system would degenerate when there are large variations in pose, expression, illumination, and face bounding box. Many robust face alignment methods prefer the end-to-end scheme, which applies abundant training data and complex network architecture to handle all the possible variations [18, 23, 38, 40]. As a result, they suffer from the low efficiency and cannot be applied on low-cost devices. Among all of these variations, the scaling and translation caused by inaccurate face detection and the in-plane rotations caused by face action can be easily normalized by simple geometric transformation. Some existing methods [14, 25, 25] apply two-stage face alignment frameworks to preliminarily normalize the geometric variations at the first stage and provide a better initialization for the second stage. However, their normalization modules only aim at the overall performance and ignore the efficiency.

In this article, we conduct an empirical analysis and observe that using normalization can not only improve the performance of face alignment but also compress calculation of the overall system. When using different frameworks with same computational cost, it is better to divide resources into a normalization network and an alignment network than to directly perform alignment with a bigger network. Specifically, our experiments show that existing normalization methods are usually suboptimal, including: using heavy networks [14], using less landmarks to normalize [10], and directly regressing transformation parameters [25]. Finally, we overcome these weaknesses and propose Fast Normalization Module (FNM) to deal with the geometric variations and make the overall framework much more efficient.

At first, FNM applies an extremely lightweight network to detect 68 coarse landmarks. Then, we apply Ordinary Procrustes Analysis to compute the best similarity transformation matrix between the detected 68 landmarks and the mean shape. Finally, we transform the input image via this similarity transform matrix. The advantages are threefold: First, the lightweight CNN of FNM greatly reduces the computational cost. Second, compared with directly predicting the parameters of similarity transformation matrix, it is better to predict the landmarks and use Ordinary Procrustes Analysis to compute the transformation matrix, because the six parameters of 2D affine transformation cannot be considered equally when training a network to regress them. Third, using

Table 1. The Network Architectures of Our Framework

Layer	FNM	LAM
Conv1.x	[5×5, 16]×1, S1	[5×5, 8]×1, S2
MaxPool	[2×2, 16], S2	[2×2, 8], S2
Conv2.x	[3×3, 24]×1, S1	[3×3, 16]×2, S1
MaxPool	[2×2, 24], S2	[2×2, 16], S2
Conv3.x	[3×3, 32]×1, S1	[3×3, 32]×2, S1
MaxPool	[2×2, 32], S2	[2×2, 32], S2
Conv4.x	[3×3, 40]×1, S1	[3×3, 64]×2, S1
FC1	64	256
FC2	$\mathrm{N} \times 2$	256
FC3	-	$\mathrm{N} \times 2$

Conv1.x, Conv2.x, and Conv3.x denote convolution units that may contain multiple convolution layers. [3×3, 16]×2 denotes 2 cascaded convolution layers with 16 filters of size 3×3 and S2 denotes stride 2. FC1, FC2, and FC3 are fully connected layers. N are the number of landmarks.

the full 68 landmarks can provide more useful supervision information to help FNM understand low-quality face images and make the normalization more stable.

4.2 Network Architecture

In Fast Normalization Module (FNM), we find the best tradeoff between performance and efficiency and design a streamlined network. We use a $30 \times 30 \times 3$ image as input and a 5-layer CNN with tiny computation. This network could conduct fitting with less than 1ms on CPU. In the experiment, we empirically try many CNN architectures with larger cost and find that this tiny CNN is powerful enough to coarsely normalize the image. Compared with previous works, the normalization network [14, 25] is much more complicated and needs about 3 to 7ms on CPU. There exists a lot of redundancy within their normalization networks. After FNM, the $120 \times 120 \times 3$ normalized image is transferred into a Lightweight Alignment Module (LAM) to predict each landmark accurately, which is trained by proposed CF Loss. The architecture of FNM and LAM is shown in Table 1. We follow the mechanism of VGGNet [34] and control the receptive field to cover the entire input image gradually. Meanwhile, we substantially reduce the channel of the two CNNs to ensure high efficiency. Extensive experiments demonstrate that our overall framework can get high performance with extremely small computation cost.

5 EXPERIMENTS

In this section, we first introduce our implementation details. Then, we analyze the proposed Contour Fitting Loss and Fast Normalization Module (FNM) to validate their effectiveness. Finally, we compare our algorithm with the state-of-the-art on the popular 300W and AFLW2000-3D datasets. Experiments demonstrate that our approach achieves competitive performance on 300W and AFLW2000-3D datasets with much lower cost.

5.1 Implementation Details

Data augmentation is important for the CNN-based face alignment task. We apply random in-plane variations (e.g., scaling, rotation, translations) to perform data augmentation. Each kind of

Table 2. A Comparison of Different Weighting Factors of the Proposed Contour Fitting Loss Function, Measured by NME(%) on 300W

λ	0	0.2	0.4	0.6	0.8	1.0
Com.	5.45	5.14	5.19	5.20	**5.09**	5.41
Challenge	8.77	8.46	8.32	8.40	**8.32**	8.50

Table 3. A comparison of NME(%) in Different Facial Areas, Including Face Contour, Facial Features, and All the Face Area

Loss Function	Contour	Facial	All
L2 Loss	12.98	7.39	8.79
CF Loss	**12.76**	**6.83**	**8.32**

augmentation follows a Gaussian distribution, as shown in Table 4. For training LAM, we apply smaller standard deviations than training FNM to ensure the CNN pays more attention to the details. During training, the basic learning rate is 0.05, and each stage is trained for 100 epochs. The learning rate is reduced by a factor of 10 after 40 and 80 epochs. We also set the momentum to 0.9, weight decay to 1×10^{-4}, and batch size to 512 for network training. SGD is used as our optimizer. Finally, we use the 300W [32] training data and do not use any extra data. We adopted the widely used Normalised Mean Error (NME) for evaluation, normalized by the inter-ocular distance.

5.2 Analysis Experiments

We use the popular 300W dataset for our evaluation in all of our experiments.

300W is a collection of multiple face datasets, including the LFPW [2], HELEN [22], AFW [27], and XM2VTS [26] face datasets that have 68 annotated landmarks [33]. The training set contains 3,148 training samples in total. The size of the test set is 689, which is further divided into the common and challenging subsets. The common subset has 554 face images, and the challenging subset constitutes the 135 IBUG face images.

AFLW2000-3D is constructed by Reference [47] to evaluate 3D face alignment on challenging unconstrained images. This database contains the first 2K images from AFLW [20] and expands its annotations with fitted 3DMM parameters and 68 3D landmarks. The NME is computed using the bounding box size. We use the projected 3D landmarks in 300W-LP [47] as our training set.

5.2.1 Analysis of the CF loss. We first evalute different hype-parameters of our proposed contour fitting loss in Table 2. First the weighting factors λ, which controls the weights of contour fitting loss. When $\lambda = 0$, our loss is equivalent to the standard L2 loss. When $\lambda = 1$, it means that we only use CF Loss for training. We observed that only using CF loss does not achieve the best results in our experiments. The reason is that the CF loss underestimates the error from contour points and hurts the performance. When $\lambda = 0.8$, our loss achieves a good balance and gets the best result.

Second, we analyze the performance promotion by the CF loss in different facial areas. We compute the NME of the contour points and the facial feature points. As shown in Table 3, the CF loss outperforms L2 loss in each evaluation criterion, especially on facial features. These results demonstrate that our CF loss alleviates the effects of regressing weak semantic points and makes CNN concentrate on learning better visual representation of other facial feature points.

Third, we design the CF Loss+, which applies point-to-line regression to all weak semantic points on the edges (e.g., nose bridge, eyebrow). As shown in Table 5, the performance of CF Loss+ degrades slightly. We think the reason for this is that the curves of other facial regions are much narrower than the face contour. Therefore, the original ground-truth points are relatively reliable. Changing the positions of these points might harm the facial structure.

Finally, to further demonstrate the effectiveness of the CF loss and observe where the improvement comes from, we decompose the error into two parts: the meaningful error perpendicular to the contour curve and the non-meaningful error parallel to the contour curve. As shown in Table 8,

Table 4. Different Standard Deviations of the Gaussians Used to Augment Training Samples for FNM and LAM

Module	Rotation	Scale	translation
FNM	17°	0.07**d**	0.08**d**
LAM	8°	0.05**d**	0.03**d**

d Is the Size of Face Boundingbox.

Table 5. A comparison of CF Loss and CF Loss+

Method	Com.	Challenge
FNM + LAM + CF loss	**5.09**	**8.32**
FNM + LAM + CF loss+	5.11	8.36
FNM + ResNet + CF loss	**4.34**	**7.40**
FNM + ResNet + CF loss+	4.39	7.47

CF Loss+ applies Point-to-line Regression to All Weak Semantic Points for Training.

Table 6. The Performance with Regressing Targets in FNM, Including Regressing 5 Landmarks, 19 Landmarks, 68 Landmarks, and Transformation Parameters

Norm Method	Com.	Challenge
5 Landmarks Norm	5.57	9.03
19 Landmarks Norm	5.47	8.98
68 Landmarks Norm	**5.44**	**8.78**
Transformation Params Norm	5.71	9.04

Table 7. Comparison of Different Input Image Sizes and Network Architectures in the Fast Normalization Module (FNM)

Size	Network	Flops(M)	Com.	Challenge
15	CNN-4	0.31	5.47	8.95
30	CNN-5	1.88	5.44	8.78
60	CNN-6	4.03	**5.43**	**8.77**
120	CNN-7	5.40	5.46	8.83

The Performance Is Evaluated on the 300W Dataset.

Table 8. A Comparison of the Performance With or Without Our CF Loss, NME of Contour Points is Decomposed Into Two Parts: The Meaningful Error Perpendicular to the Contour Curve (Prep NME) and the Non-meaningful Error Parallel to the Contour Curve (Para NME)

Dataset	Method	Prep NME	Para NME	NME
Com.	LAM	4.38	5.50	8.04
Com.	LAM + CF Loss	**4.01(↓ 8.4%)**	**5.37(↓ 2.3%)**	**7.63**
IBUG	LAM	7.53	9.11	13.89
IBUG	LAM + CF Loss	**6.84(↓ 9.2%)**	**8.98(↓ 1.4%)**	**13.19**

when we replace the L2 loss with our CF loss, the error perpendicular to the contour curve decreases more significantly than the error parallel to the contour curve (8.4% vs 2.3% on Common and 9.2% vs 1.4% on IBUG). These results demonstrate that most of the performance promotion comes from the decrease of point-to-line error.

5.2.2 Analysis of Fast Normalization Module. The proposed Fast Normalization Module (FNM) efficiently normalizes the scaling, translation, and in-plane rotations in our face alignment system. To reduce the computation time in our FNM, we first design a series of CNNs with different sizes and different input sizes and evaluate their performance. As shown in Table 7, we try four kinds of network architecture and find that a 30 × 30 input image with a five-layer CNN can well preserve the face structure and get competitive results with larger networks. When we reduce the input size to 15 × 15, the effectiveness of normalization starts to degrade. To balance efficiency and performance, we employ 30 × 30 input images in our FNM. We further investigate the performance with different kinds of geometric transformations, as shown in Table 9. The deformable affine transformation in Reference [14] performs worse than the non-deformable similarity transformaton used in our method.

In addition, we use different regression targets in the Fast Normalization Module (FNM) and explore their influences to the final results. We test a different number of landmarks and the

Table 9. The Performance of Different Transformation Methods

Transform	Com.	Challenge
Affine	5.83	9.61
Similarity	**5.44**	**8.78**

We Test the Deformable Affine Transformation and the Non-deformable Similarity Transformation.

Table 10. The Effectiveness of Normalization

Norm Method	Com.	Challenge
LAM+	**5.40**	9.37
LAM with gt Norm	5.47	**8.75**
LAM with FNM	5.45	8.78

We Evaluate the Performance of the Heavier CNN LAM+, Normalization with Ground-truth Points and With FNM.

Table 11. Ablation Study of Our Overall Framework, Evaluated by the NME(%)

Method	Com.	Challenge
LAM + L2 Loss	5.46	9.49
LAM + CF Loss	5.20	8.89
FNM + LAM + L2 Loss	5.45	8.79
FNM + LAM + CF Loss	**5.09**	**8.32**

parameters of transformation matrix shown in Table 6. In this experiment, we find regressing transformation matrix parameters gets the worst performance. This may be explained by the fact that the six parameters of 2D affine transformation cannot be considered equally in the normalization process. When we regress the landmarks in FNM, the results show that using all 68 points performs better than using part of them. We believe the reason of this experimental phenomena is that FNM applies extremely low-resolution images (30×30) as its input, which needs more supervision information to be understood. Meanwhile, because FNM applies similarity transformation to normalize in-plane variations, more landmarks can also benefit the optimization of least squares method. Therefore, training FNM with a smaller number of landmarks would harm the performance of normalization.

Finally, we evaluate the overall normalization performance in FNM. We use the ground-truth points and the predicted points by our FNM to normalize the geometric variations. As shown in Table 10, our proposed FNM achieves competitive performance with the ground-truth normalization. We also design a heavier CNN named LAM+ by adding channel dimensions. LAM+ has equal computation cost with the combination of FMN and LAM but performs worse on the challenge dataset. This result demonstrates the effectiveness of our FNM.

5.3 Ablation Experiments

To demonstrate the effectiveness of different components in our proposed framework, we run a number of ablations to analyze the contour fitting loss and the Fast Normalization Module (FNM), shown in Table 11. Specifically, for a fair comparison, we use the same parameter settings in all these experiments. We first analyze the performance and find that the FNM significantly improves the performance on the challenge dataset with large pose variations. There are no obvious improvements on the common dataset because of its small pose variations. These results demonstrate the effectiveness of the FNM on normalizing geometric variances. To further validate the effectiveness of our CF loss, we use the original training images and the normalized images to train with CF loss. Results show that the CF loss consistently outperforms the L2 loss at each dataset. Finally, the combination of CF-Loss and FNM provide the best performance.

Table 12. A Comparison of the Proposed Approach with the State-of-the-art Approaches on the 300W Dataset in Terms of the NME Averaged Over All the Test Samples

Method	Com.(%)	Challenge(%)	Full	Flops(M)	Params(M)	CPU Time(ms)
RCPR [5]	6.18	17.26	8.35	-	-	-
CFAN [42]	5.50	16.78	7.69	-	-	-
ESR [6]	5.28	17.00	7.58	-	-	-
SDM [39]	5.60	15.40	7.52	-	-	-
ERT [19]	-	-	6.40	-	-	-
CFSS [46]	4.73	9.98	5.76	-	-	-
TCDCN [45]	4.80	8.60	5.54	22.02	0.32	3.06
LBF [30]	4.95	11.98	6.32	-	-	-
3DDFA [47]	6.15	10.59	7.01	41.75	1.44	7.68
3DDFA + SDM	5.53	9.56	6.31	-	-	-
DDN [41]	-	-	5.65	1,779.84	33.47	74.77
RAR [38]	4.12	8.35	4.94	-	-	-
DeFA [23]	5.37	9.38	6.10	1,425.57	3.18	97.70
TR-DRN [25]	4.36	7.56	4.99	2,658.04	99.65	126.52
PCD-CNN* [21]	3.67	7.62	4.44	10K+	-	-
LAB* [37]	**3.42**	**6.98**	4.12	10K+	-	-
DCFE* [36]	3.83	7.54	4.55	10K+	-	-
LAM	7.40	11.90	6.16	5.97	0.19	2.42
FNM + LAM	5.45	8.79	6.10	**7.85**	**0.22**	2.42
FNM + LAM + CF Loss($\lambda = 0.8$)	**5.09**	**8.32**	**5.72**	**7.85**	**0.22**	2.42
ResNet	5.19	8.35	5.81	1,746.38	11.69	103.00
FNM + ResNet	4.87	8.08	5.50	1,748.23	11.73	104.00
FNM + ResNet + CF Loss($\lambda = 0.8$)	**4.34**	**7.40**	**4.93**	1,748.23	11.73	104.00

We Follow the Protocol Used in Reference [30]. *Represents the Heatmap-based Methods.

5.4 Comparison Experiments

300W. We compare the performance and computational complexity of our method with other state-of-the-art methods on the famous 300W dataset in Table 12. Specifically, we apply different backbones such as lightweight LAM and the deeper 25-layer ResNet [15] to validate the effectiveness of our common efficient face alignment framework. As shown in Table 12, we can see our Fast Normalization Module (FNM) brings significant improvement (6.10% vs 5.72% on LAM; 5.81% vs 5.50% on ResNet) with nearly free computing overhead. This result demonstrates that our high-speed FNM with extremely low-resolution input can still perform well in normalizing the complex geometric variations of different human faces. In addition, compared with FNM + LAM and FNM + ResNet, replacing the common point-to-point loss to our contour fitting loss (CF loss) further reduces the NME by a significant margin (6.10% vs 5.72% on LAM; 5.50% vs 4.93% on ResNet). This achievement means that our CF loss alleviates the impact of potential annotation randomness caused by contour points and makes CNN pay more attention to the better optimization direction. The above analysis proves that our FNM and CF loss both have a strong generalized ability on different backbones such as LAM and ResNet.

Compared with other existing methods, heatmap-based methods such as PCD-CNN [21], LAB [37], and DCFE [36] have better performance; however, they tend to be computationally prohibitive and cannot be applied on CPU-only devices (over 10G flops). Our FNM + ResNet + CF Loss outperforms all of the state-of-the-art methods except the heatmap-based methods.

Table 13. Performance Comparison on AFLW2000-3D Dataset

Method	[0°30°]	[30°60°]	[60°90°]	Mean
3DSTN [4]	3.15	4.33	5.98	4.49
Hyperface [28]	3.93	4.14	4.71	4.26
3DDFA [48]	2.84	3.57	4.96	3.79
PRN [13]	2.75	3.51	4.61	3.62
LAM	3.70	4.71	5.38	4.60
FNM + LAM	3.20	3.89	4.93	4.00
FNM + LAM + CF Loss	**2.98**	**3.77**	**4.68**	**3.81**
ResNet	3.01	3.74	4.81	3.85
FNM + ResNet	2.57	3.30	4.33	3.40
FNM + ResNet + CF Loss	**2.56**	**3.22**	**4.17**	**3.31**

The NME (%) for Faces with Different Yaw Angles Are Reported.

Meanwhile, our fast-speed framework FNM + LAM + CF Loss has a speed of 340FPS on a single core i7-6700 CPU and still achieves competitive performance with other methods, even better than using ResNet only (5.72% vs 5.81%). These results demonstrate that our method can well benefit the deployment of face alignment framework on platforms with limited computational resources. In addition, our FNM and CF loss can team with any backbone according to the requirements of applications. Therefore, We believe that we can get further performance promotion when using a more complex backbone.

AFLW2000-3D. To demonstrate the effectiveness of our framework on addressing the large pose variations. We evaluate our method on AFLW2000-3D [47] dataset. We use the projected 68 3D landmarks from 300W-LP [47] as the ground truth to train our method. As shown in Table 13, our framework with a simple 25-layer ResNet [15] outperforms other state-of-the-art methods. We can observe that our FNM achieves significant improvement on both LAM and ResNet structures (Mean NME: 3.81% vs 4.00% on LAM and 3.40% vs 3.85% on ResNet). This result demonstrates the effectiveness of our FNM on the dataset with large pose variations. In addition, using our CF loss further improves the overall performance.

6 CONCLUSIONS

In this article, we concentrate on exploring a CNN-based face alignment framework with both high efficiency and performance. First, we analyze the different semantic meanings of facial landmarks and propose the concept of strong and weak semantic point. Due to the potential randomness of the annotation of weak semantic points, we propose a novel contour fitting loss to help the network better understand the semantic positions of weak semantic points. Contour fitting loss effectively improves the overall performance. Meanwhile, we systematically analyze the different pose variations in face alignment task and design an efficient Fast Normalization Module (FNM) to normalize geometric variations, which is simple to solve but crucial for facial landmark localization. Our method achieves competitive performance with the state-of-the-art methods on 300W and AFLW2000-3D benchmarks with significantly lower computation cost.

REFERENCES

[1] Brian Amberg, Sami Romdhani, and Thomas Vetter. 2007. Optimal step nonrigid ICP algorithms for surface registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1–8.

[2] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. 2011. Localizing parts of faces using a consensus of exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 545–552.

[3] Paul J. Besl and Neil D. McKay. 1992. Method for registration of 3-D shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, Vol. 1611. International Society for Optics and Photonics, 586–607.

[4] Chandrasekhar Bhagavatula, Chenchen Zhu, Khoa Luu, and Marios Savvides. 2017. Faster than real-time facial alignment: A 3D spatial transformer network approach in unconstrained poses. In *Proceedings of the IEEE International Conference on Computer Vision*. 3980–3989.

[5] Xavier P. Burgos-Artizzu, Pietro Perona, and Piotr Dollár. 2013. Robust face landmark estimation under occlusion. In *Proceedings of the IEEE International Conference on Computer Vision*. 1513–1520.

[6] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. 2014. Face alignment by explicit shape regression. *Int. J. Comput. Vis.* 107, 2 (2014), 177–190.

[7] T. F. Cootes, G. J. Edwards, and C. J. Taylor. 1998. Active appearance models. In *European Conference on Computer Vision*. 484–498.

[8] Timothy F. Cootes, Christopher J. Taylor, David H. Cooper, and Jim Graham. 1995. Active shape models-their training and application. *Computer Vision and Image Understanding* 61, 1 (1995), 38–59.

[9] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. *arXiv preprint arXiv:1602.02830* (2016).

[10] Jiankang Deng, George Trigeorgis, Yuxiang Zhou, and Stefanos Zafeiriou. 2019. Joint multi-view face alignment in the wild. *IEEE Transactions on Image Processing* 28, 7 (2019), 3636–3648.

[11] P. Dollár, P. Welinder, and P. Perona. 2010. Cascaded pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1078–1085.

[12] Xuanyi Dong, Shoou-I Yu, Xinshuo Weng, Shih-En Wei, Yi Yang, and Yaser Sheikh. 2018. Supervision-by-registration: An unsupervised approach to improve the precision of facial landmark detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 360–368.

[13] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. 2018. Joint 3D face reconstruction and dense alignment with position map regression network. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*. 534–551.

[14] Zhen Hua Feng, Josef Kittler, Muhammad Awais, Patrik Huber, and Xiao Jun Wu. 2018. Wing loss for robust facial landmark localisation with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2235–2245.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[16] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[17] Gao Huang, Shichen Liu, Van Der Maaten Laurens, and Kilian Q. Weinberger. 2018. CondenseNet: An efficient densenet using learned group convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2752–2761.

[18] Amin Jourabloo, Mao Ye, Xiaoming Liu, and Liu Ren. 2017. Pose-invariant face alignment with a single CNN. In *Proceedings of the IEEE International Conference on Computer Vision*. 3219–3228.

[19] Vahid Kazemi and Josephine Sullivan. 2014. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1867–1874.

[20] Martin Koestinger, Paul Wohlhart, Peter M. Roth, and Horst Bischof. 2011. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops'11)*. IEEE, 2144–2151.

[21] Amit Kumar and Rama Chellappa. 2018. Disentangling 3D pose in a dendritic CNN for unconstrained 2D face alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 430–439.

[22] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir Bourdev, and Thomas S. Huang. 2012. Interactive facial feature localization. In *Proceedings of the European Conference on Computer Vision*. 679–692.

[23] Yaojie Liu, Amin Jourabloo, William Ren, and Xiaoming Liu. 2017. Dense face alignment. In *Proceedings of the IEEE International Conference on Computer Vision Workshop*. 1619–1628.

[24] David G. Lowe. 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 60, 2 (2004), 91–110.

[25] Jiangjing Lv, Xiaohu Shao, Junliang Xing, Cheng Cheng, and Xi Zhou. 2017. A deep regression architecture with two-stage re-initialization for high performance facial landmark detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3691–3700.

[26] K. Messer, J. Matas, J. Kittler, and K. Jonsson. 2000. XM2VTS: the extended M2VTS database. In *Proceedings of the 2nd International Conference on Audio- and Video-Based Biometric Person Authentication*. 72–77.

[27] Deva Ramanan. 2012. Face detection, pose estimation, and landmark localization in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2879–2886.

[28] Rajeev Ranjan, Vishal M. Patel, and Rama Chellappa. 2019. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Trans. Pattern Anal. Machine Intell.* 41, 1 (2019), 121–135.

[29] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. 2016. XNOR-Net: ImageNet classification using binary convolutional neural networks. In *European Conference on Computer Vision.* Springer, 525–542.

[30] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. 2016. Face alignment via regressing local binary features. *IEEE Trans. Image Proc.* 25, 3 (2016), 1233–1245.

[31] Sami Romdhani and Thomas Vetter. 2005. Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 2. IEEE, 986–993.

[32] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 2013. 300 Faces in-the-wild challenge: The first facial landmark localization challenge. In *Proceedings of the IEEE International Conference on Computer Vision Workshops.* 397–403.

[33] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 2013. A semi-automatic methodology for facial landmark annotation. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Workshops.* 896–903.

[34] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[35] Yi Sun, Xiaogang Wang, and Xiaoou Tang. 2013. Deep convolutional network cascade for facial point detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 3476–3483.

[36] Roberto Valle and M. José. 2018. A deeply initialized coarse-to-fine ensemble of regression trees for face alignment. In *Proceedings of the European Conference on Computer Vision (ECCV'18).* 585–601.

[37] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. 2018. Look at boundary: A boundary-aware face alignment algorithm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2129–2138.

[38] Shengtao Xiao, Jiashi Feng, Junliang Xing, Hanjiang Lai, Shuicheng Yan, and Ashraf Kassim. 2016. Robust facial landmark detection via recurrent attentive-refinement networks. In *Proceedings of the European Conference on Computer Vision.* 57–72.

[39] Xuehan Xiong and Fernando De La Torre. 2013. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 532–539.

[40] Jing Yang, Qingshan Liu, and Kaihua Zhang. 2017. Stacked hourglass network for robust facial landmark localisation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops.* 2025–2033.

[41] Xiang Yu, Feng Zhou, and Manmohan Chandraker. 2016. Deep deformation network for object landmark localization. In *Proceedings of the European Conference on Computer Vision* (2016), 52–70.

[42] Jie Zhang, Shiguang Shan, Meina Kan, and Xilin Chen. 2014. Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment. In *Proceedings of the European Conference on Computer Vision.* 1–16.

[43] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Sig. Proc. Lett.* 23, 10 (2016), 1499–1503.

[44] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. 2018. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 6848–6856.

[45] Zhanpeng Zhang, Ping Luo, Change Loy Chen, and Xiaoou Tang. 2014. Facial landmark detection by deep multi-task learning. In *Proceedings of the European Conference on Computer Vision.* 94–108.

[46] Shizhan Zhu, Cheng Li, Change Loy Chen, and Xiaoou Tang. 2015. Face alignment by coarse-to-fine shape searching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 4998–5006.

[47] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li. 2016. Face alignment across large poses: A 3D solution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 146–155.

[48] Xiangyu Zhu, Xiaoming Liu, Zhen Lei, and Stan Z. Li. 2019. Face alignment in full pose range: A 3D total solution. *IEEE Trans. Pattern Anal. Machine Intell.* 41, 1 (2019), 78–92.

[49] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. 2018. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 8697–8710.

Received October 2018; revised March 2019; accepted May 2019