Robust Face Alignment via Deep Progressive Reinitialization and Adaptive Error-driven Learning

Xiaohu Shao*, Junliang Xing*, *Senior Member, IEEE,* Jiangjing Lyu, Xiangdong Zhou, Yu Shi, and Steve Maybank, *Fellow, IEEE*

Abstract—Regression-based face alignment involves learning a series of mapping functions to predict the true landmark from an initial estimation of the alignment. Most existing approaches focus on learning efficacious mapping functions from some feature representations to improve performance. The issues related to the initial alignment estimation and the final learning objective, however, receive less attention. This work proposes a deep regression architecture with progressive reinitialization and a new error-driven learning loss function to explicitly address the above two issues. Given an image with a rough face detection result, the full face region is firstly mapped by a supervised spatial transformer network to a normalized form and trained to regress coarse positions of landmarks. Then, different face parts are further respectively reinitialized to their own normalized states, followed by another regression sub-network to refine the landmark positions. To deal with the inconsistent annotations in existing training datasets, we further propose an adaptive landmark-weighted loss function. It dynamically adjusts the importance of different landmarks according to their learning errors during training without depending on any hyper-parameters manually set by trial and error. A high level of robustness to annotation inconsistencies is thus achieved. The whole deep architecture permits training from end to end, and extensive experimental analyses and comparisons demonstrate its effectiveness and efficiency. We will release the source code, trained models, and experimental results upon the publication of this work.

Index Terms—Face Alignment, Regression Model, Deep Architecture, Supervised Spatial Transformer Network, Adaptive Learning

1 Introduction

FACE alignment involves locating predefined landmarks on a face image, usually obtained by a face detector. It is an essential and fundamental task in computer vision and acts as a critical component of many other tasks, *e.g.*, face tracking [1], [2], face animation [3], [4], and face recognition [5], [6]. Despite significant progress in the past decades [1], [7]–[15], face alignment remains a very challenging problem, especially when face images show large head pose variations, exaggerated facial expressions, and partial face occlusions.

Regression-based face alignment methods [1], [10], [13], [14], [16]–[23] are currently dominant. They directly learn a series of mapping functions, *i.e.*, regressors, using some image features to iteratively update the estimates of landmark positions to converge to true values, starting from an initial estimate. Typical regression algorithms tailored for face alignment include random ferns [10], [13], [16], least squares regression [1], [21], random forests [18], and support vector regression [11], [24]. For image features, classical image descriptors (*e.g.*, SIFT [1], [21] and HoG [14], [25]) and simple ones (*e.g.*, pixel differences [10], [13], [16] and local binary features [17]) are widely used in existing methods.

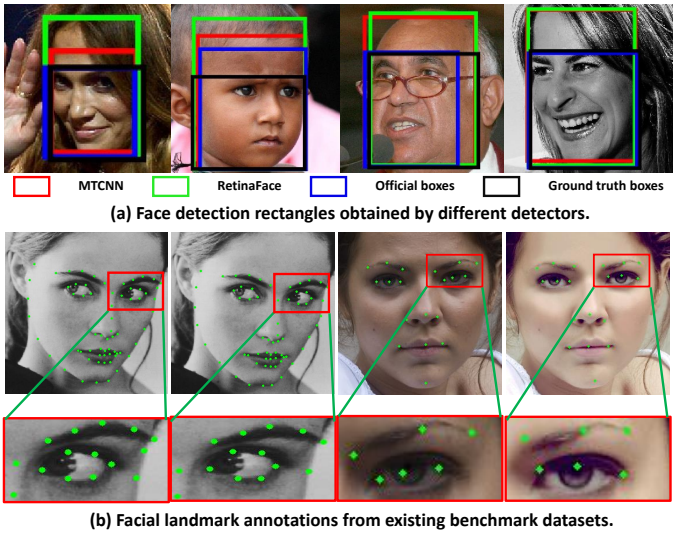


Fig. 1. Training a system for face alignment is difficult because of the significant differences of results from face detectors (a) and the notable variations in ground truth landmark annotations (b).

Recently, with the fast deployment of deep learning-based face alignment models, feature representation and regression learning are incorporated together into one framework using Convolutional Neural Networks (CNN) [15],

- * *indicates equal contributions*
- *X. Shao, Xiangdong Zhou, Y. Shi and Xi Zhou are with Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences. X. Shao is also with the University of Chinese Academy of Sciences. E-mail: {shaoxiaohu, zhouxiangdong, shiyu}@cigit.ac.cn*
- *J. Xing is with the Institute of Automation, Chinese Academy of Sciences. E-mail: jlxing@nlpr.ia.ac.cn*
- *J. Lyu is with Alibaba Group. E-mail: jiangjing.ljj@alibaba-inc.com*
- *S. Maybank is with the Department of Computer Science and Information Systems, Birkbeck College, University of London. E-mail: sjmaybank@dcs.bbk.ac.uk*

[26] or Recurrent Neural Networks [27]. These deep learning-based algorithms have significantly improved the alignment of near-frontal faces. However, for face images with significant view variations, different expressions, and partial occlusions, even state-of-the-art algorithms may still fail to locate the landmarks correctly. These deficiencies restrict the applications of exiting face alignment algorithms in practical systems. To deal with these challenges, previous works [12], [19]–[22], [28] focus on learning robust image features and sophisticated regression functions to improve the performance of face alignment. However, the learning initialization and learning objective of a deep face alignment model have received much less attention.

The Learning Initialization Issue Most existing algorithms depend heavily on face detection to provide a good rectangular face region as an initialization. According to recent studies [29]–[31], if a face alignment model in the testing phase uses face detection produced by a different algorithm from that used in training, the alignment accuracy will be degraded. In many situations, users may have to choose a new face detector if the one used in the training process is not available. Since different face detectors provide face bounding boxes with different scales and center shifts (*c.f.* Fig. 1 (a)), they impose great difficulties to face alignment for face shape modeling and learning.

The Learning Objective Issue Regression-based face alignment algorithms learn to predict the landmark locations provided by some human annotators in the benchmarks. Since there is no uniform annotation protocol for landmark positions, these landmark annotations usually exhibit significant variations in different benchmarks. As shown in Fig. 1 (b), the landmark annotations for two almost identical face images are not consistent. This situation is conspicuously worse for some specific landmarks, *e.g.*, landmarks on face contour or nose bridge. These observations reveal that the facial landmark annotations in reality are often imperfect, and designing a face alignment model to alleviate the effects of inconsistent annotations is necessary.

To deal with the above two issues, we present a novel deep architecture with progressive reinitialization and adaptive error-driven learning to obtain high-performance face alignment results. The proposed model first reinitialize the whole face region spatially into a normalized state for better landmark estimation. Then it further reinitializes different facial parts into their normalized states to deal with expression variations and partial face occlusions. To take account of the prediction errors and the unreliability of labels for different landmarks, we design an adaptive landmark-weighted loss function to dynamically adjust the annotation reliability of different landmarks according to their learning errors during the training procedure. The resulting deep architecture permits training from end to end and produces accurate and robust facial alignments.

An early version of this work appeared in the conference paper [32]. We extend it in numerous ways, (i) exploiting a novel landmark-weighted loss function for error-driven learning, (ii) generalizing the progressive reinitialization, and (iii) the error-driven learning to a series of well-designed backbones with better accuracy and efficiency. To summarize, the main contributions of this work are as follows:

- A deep regression architecture with progressive reinitialization and error-driven learning is proposed to solve the initialization problems and labeling inconsistency for robust face alignment.
- A progressive reinitialization procedure is formulated as a supervised spatial transformer learning problem that leverages both global face shape and local face parts to make the alignment model invariant to different face detection inputs.
- An adaptive landmark-weighted loss function, which have no hyper-parameters manually set by trial and error, is introduced for error-driven regression learning to reduce the adverse impact of the inconsistencies in manual landmark annotation.
- The good generalization of our proposed method to different backbones helps to find a better trade-off between the model accuracy and efficiency, especially in real-time practice applications without GPU support.

With the above technical contributions, we obtain a fast, accurate, and robust deep regression-based face alignment model. Extensive experimental analyses demonstrate the robustness of the proposed method to different kinds of initialization and facial landmark annotations. On four of the widely adopted face alignment benchmarks, the proposed model consistently achieves superior performances over many competing algorithms in terms of accuracy and efficiency. To facilitate further studies on the face alignment problem, the source code, trained models, and all the experimental results will be released upon the paper publication.

2 Related Work

The face alignment problem has been studied for decades [1], [12], [14]. Recently deep learning based methods [15], [21], [22], [26]–[28], [32]–[42] have consistently improved the alignment performances. Since hundreds of papers have been published in the last decade, we only discuss those closely related to this work.

Joint Learning Methods These works view face alignment and face detection as two correlated tasks and leverage the correlations by learning them jointly to boost the performance. The joint cascade face detection and alignment method [43] formulates face alignment as a post-classification task, and reduces the false alarm rate of the face detector. MTCNN [44] regards the two tasks as equally important. They share most of the neural layers. The locations of a face bounding box and five sparse landmarks are predicted simultaneously. Besides face detection, many other face attributes, such as age, expression, gender, and identity, have also been explored to improve face alignment models using multi-task learning [45]–[47]. Although the multi-task joint learning improves each module's accuracy, face detection and alignment still operate in two independent branches. Face alignment is inevitably affected by the location instabilities in face detection.

Initialization Optimization These works focus on optimizing the initialization provided by face detectors before landmark regression. Some works [2], [21], [30] optimize the initialization of facial shapes at different regression stages, other methods transform the face images progressively to update results. The coarse-to-fine cascade model [48] combines multi-level geometric refinement to rectify facial

landmarks. Another face alignment model [49] uses multi-scale local image patches to perform cascade regression on a normalized full-face image. The recent work [47] firstly locate sparse landmarks and then feed them into the spatial transformer in the subsequent alignment network. In contrast with these methods of employing hand-designed sample transformations to learn a series of regression functions or networks, our model is an end-to-end architecture by automatically building the reinitialization modules across the training information's priors.

DAN [50] achieves high performance with only a rough initialization by deploying heat maps of landmarks as prior visual information. The boundary-aware LAB [36] introduces a fusion scheme of the boundary heat map to incorporate boundary information into the feature learning stage. DeCaFA [51] uses fully-convolutional stages with chained transfer layers to produce landmark-wise attention maps for landmark regression. KDN [52] introduces a kernel density deep neural network with a landmark probability map as its output. Heat maps of landmarks or boundaries improve the initialization for face alignment, but their portability and application range are limited by high computational costs.

Deal with Annotation Inconsistency Only a few works deal with the annotation inconsistency when training a neural network for face alignment. The supervision-by-registration approach [53] augments the loss function with registration information automatically extracted from unlabeled data and thus reduces the dependence on manual annotation of video-based face alignment. SAN [34] utilizes the detection inconsistency on style-aggregated images generated by an adversarial module to enhance the robustness of face alignment on face images with various styles. The semantic alignment model [38] introduces a probabilistic model to search for the 'real' ground-truth and train the face alignment model. These approaches utilize additional complex information, *e.g.*, optical flow, different styles, or face shape constraints, to optimize the training annotations and predicted results. In contrast, we present a simple and effective solution to reduce the impact of unreliable annotations and achieve competitive performance.

Different Learning Objectives Several variations of the ℓ_1 or ℓ_2 based loss functions designed for face alignment have exhibited excellent accuracy. Wing [35] and RWing [54] both focus on small range errors and switch the loss function from an ℓ_1 loss to a modified logarithm function. AWing [55] applies a similar idea to improve the quality of heat map regression results. LUVLi [56] jointly estimates landmark locations, uncertainties, and visibilities using the spatial mean of the positive elements of each landmark heat map. In contrast with these methods using empirically specified parameters or heat maps, we introduce adaptive self-learning weights, automatically driven by location errors during the training procedure, to enhance the regression learning of all landmarks. This loss function aims to avoid over-fitting of regression models to inconsistent annotations of the training samples.

3 Our Method

Given an image, the objective of face alignment is to locate the face shape $\mathbf{S}$ specified by the positions of n facial landmarks, *i.e.*, $\mathbf{S} = (\mathbf{p}_1, \ldots, \mathbf{p}_n) \in \mathbb{R}^{2\times n}$, where $\mathbf{p}_i = (x, y)^\top$ is the two-dimensional coordinate of the i-th landmark, $i \in \{1, \ldots, n\}$. Regression-based face alignment algorithms achieve this objective by learning a regression function from an original cropped face image F_0, normally obtained from a face detector. For our method, the instability in the face detector's output is overcome by reinitializing the face shape estimation and refining the final estimate of the face alignment, working from global to local. Therefore, in the proposed regression model, the face shape estimation is varying and updated at different training stages. For convenience in the following description, the subscripts $\cdot_g$ and $\cdot_l$ represent the variables at the global and local stage, the superscripts $\cdot^*$ and $\hat{\cdot}$ represent the ground truth and predicted results, *e.g.*, $\mathbf{S}_g^*$ denotes the ground truth shapes at the global stage.

In the following, we first overview the pipeline of the proposed two-stage reinitialization deep architecture. Then we introduce its global and local shape regression stages. Finally, we discuss the annotation inconsistency problem and present the new landmark-weighted loss function.

3.1 Architecture overview

Fig. 2 shows the pipeline of the proposed deep regression architecture with two-stage reinitialization for coarse-to-fine facial landmark detection. It consists of two stages, *Global Shape Regression* (GSR) and *Local Shape Regression* (LSR), both of which are learned by an adaptive landmark-weighted loss function for error-driven learning. The whole deep architecture successively reinitializes a deep regression model from coarse to fine, and global to local, to boost face alignment performance. Although the whole deep architecture contains multiple sub-networks, it permits end-to-end training since all the sub-networks are seamlessly concatenated.

Given an input image I with a bounding box R, the GSR stage sends the coarse cropped face image F_0 into its reinitialization sub-network, which spatially transforms it into a normalized state[1] to obtain the global reinitialized face F_g. Then the global regression sub-network learns to obtain the global face shape $\hat{\mathbf{S}}_g$. The LSR stage splits the whole face shape into five face parts $\{\hat{\mathbf{S}}_g^1, \ldots, \hat{\mathbf{S}}_g^5\}$, each of which is independently reinitialized to its own normalized state in the local reinitialization sub-network and then further updated by the local regression sub-network using the adaptive weighted loss. The local face shape $\hat{\mathbf{S}}_l$ obtained by the LSR stage is projected back on I to obtain the final face shape $\hat{\mathbf{S}}$.

3.2 Global Shape Regression (GSR)

The GSR stage aims to refine the rough face bounding box and learn a better initialization of the input face region. In contrast to most works which predict the landmark locations from the original face F_0 directly, this stage firstly learns to normalize F_0 to F_g and then feed F_g into the following regression sub-network for predicting the global face shape $\hat{\mathbf{S}}_g$.

Global Reinitialization Sub-network The face box bounded by the ground truth shape often works as the best

1. In the experiments, we adopt the centralized upright face as the normalized state, following common practice.

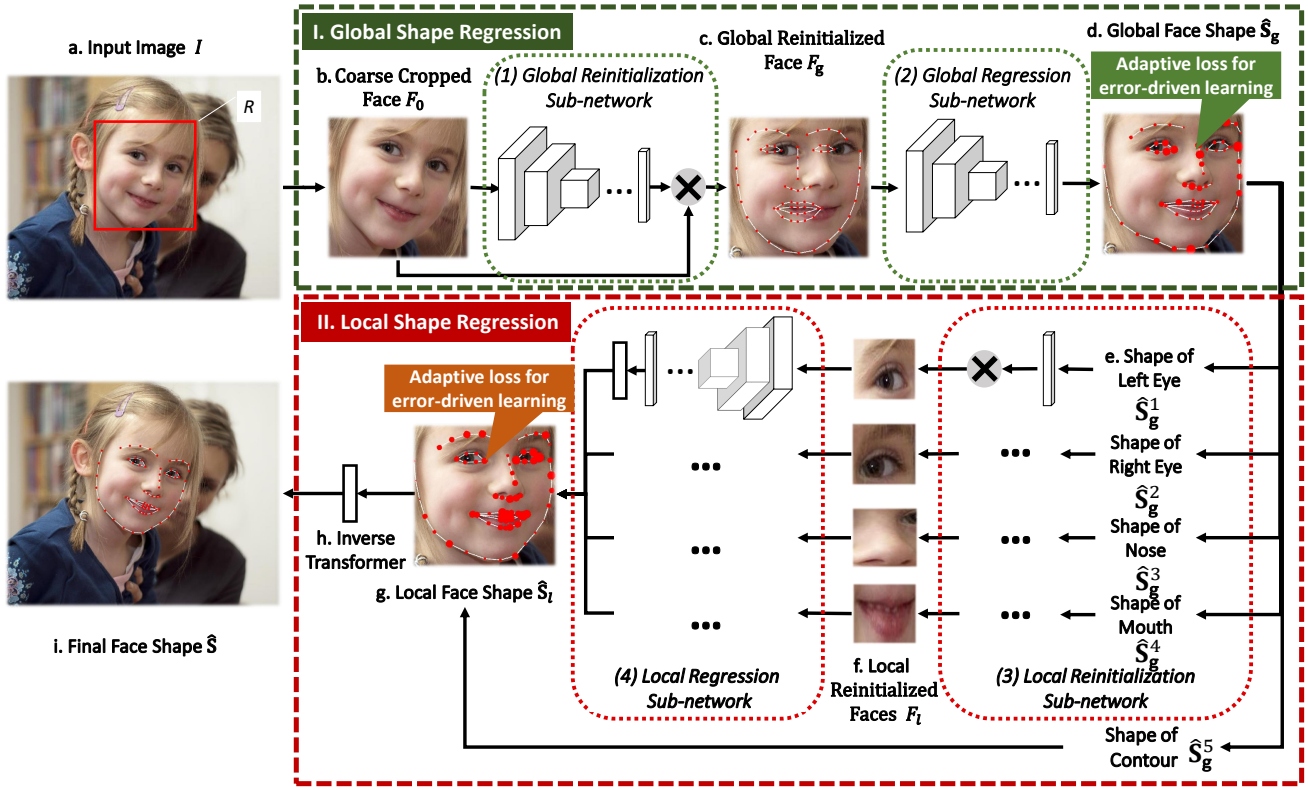


Fig. 2. The pipeline of the proposed deep regression architecture with two-stage reinitialization and error-driven learning. At the GSR stage (I), the coarse face (b) is firstly reinitialized to a normalized shape state (c), and then regresses a rough face shape (e). At the LSR stage (II), different face parts (f) are further separately reinitialized to their normalized shape states (g), followed by local regression sub-networks to get the final detection (h). The final shape is projected back to the initial coordinate (j). All landmarks in (e) and (h) are drawn by red circles with different sizes, representing their weight values of the landmark-weighted loss function regression in training progress.

initialization for the regression of face alignment in previous studies [29], [31]. The ground truth face box, however, is unknown in the testing phase. This global reinitialization sub-network can reinitialize F_0 into F_g to alleviate the appearance variations in the output of face detectors. It is built upon the Spatial Transformer Network (STN) [57]. Here we thus give a short review of STN first. The STN can produce an appropriate geometric transformation on its input face image for the follow-up task. An STN consists of three modules: 1) a localization network, which aims to predict the spatial transformation parameters; 2) a grid generator to create a sampling grid for an image, which produces the transformed image; 3) a sampler, which takes the input image and the grid to produce the transformed image.

We employ an affine transformation[2] as the learning target for the localization network. The norm form F_g is obtained by translating the initial face region of F_0 to the image center, rotating the face to the upright viewpoint with skew deformations, resizing the face to the fixed size and cutting out unnecessary background. A lightweight CNN structure, whose detailed configuration is depicted in Fig. 3, is designed to learn the transformation parameter θ_g. θ_g consists of six elements that makes up a transformation matrix $T_{\theta_g} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix}$ for further image transformation.

2. Other geometric transformer functions, *e.g.*, similarity transformation, and prospective transformation, are also feasible for the state reinitialization.

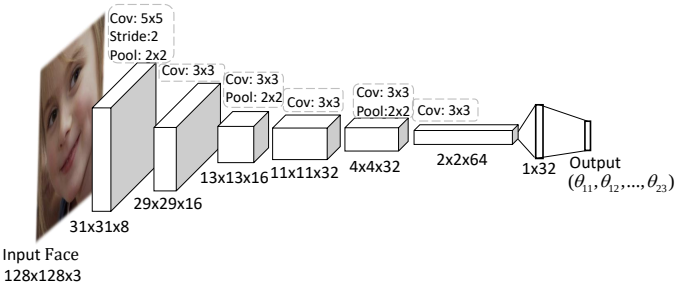


Fig. 3. Detailed configuration of the localization network designed for the global reinitialization module.

The sample point (x_i^s, y_i^s) on F_0 can be formed by using T_{θ_g} and the target point (x_i^t, y_i^t) of the regular grid on F_g:

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = T_{\theta_g} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}. \quad (1)$$

In the sampler, each pixel value of F_g is bilinearly interpolated from the corresponding pixels of F_0:

$$F_g = \Psi(F_0, T_{\theta_g}), \quad (2)$$

where Ψ represents the bilinear sampler [57]. The grid generator and sampler are both differentiable, allowing gradients to be backpropagated through from the sampler $\Psi(F_0, T_{\theta_g})$ to θ_g.

In the original STN model for handwriting digit recognition, the transformation parameters are learned from the gradients back-propagated from the final classification

loss [57]. For the task of face alignment on various face images, the STN model followed by a traditional CNN-based landmark regressor converges slowly in training. Different from [58], [59] using pixel-wise intensity similarity as STN's constraint on the training of their up-sampling networks, for our lightweight down-sampling CNN structure, a supervised STN with a novel loss function $\mathcal{L}_{\theta_g}$ for directly learning the low-dimensional parameter θ_g is introduced to accelerate the convergence speed. The loss function $\mathcal{L}_{\theta_g}$ writes as:

$$\mathcal{L}_{\theta_g} = ||\hat{\theta}_g - \theta_g^*||_2^2, \tag{3}$$

where θ_g^* is an interim target for θ_g, the detailed access to obtain θ_g^* is described in Algorithm 1. When F_g is spatial transformed from F_0 by Eq. 2, its corresponding ground truth shape $\mathbf{S}_g^*$ needs also be mapped from the original coordinate space of F_0 to that of F_g, as the target of the subsequent shape regression,

$$\mathbf{S}_g^* = T_{\theta_g}^{-1}\begin{pmatrix} \mathbf{S}^* \\ 1 \end{pmatrix}, \tag{4}$$

where $\mathbf{S}^*$ denotes the original ground truth shape, and $T_{\theta_g}^{-1}$ is the inverse transformation to T_{θ_g}.

Algorithm 1 The method of obtaining θ_g^*.

Input: A reference frontal mean shape $\mathbf{S}_{\text{ref}}$, the coarse cropped face F_0 and its ground truth shape $\mathbf{S}^*$.
Output: Affine transformation parameter θ_g^*.
1: Construct the intermediate affine transformation parameter θ_{int}, which maps the source shape $\mathbf{S}^*$ to the target shape $\mathbf{S}_{\text{ref}}$.
2: F_0 is warped to the intermediate normalized face image F_1 with the warped ground truth shape $\mathbf{S}_1^*$ by adopting θ_{int}.
3: The face regions bounding by $\mathbf{S}_1^*$ on F_1 are cropped and resized as the final normalized image F_2. $\mathbf{S}_1^*$ is also mapped to the coordinate space of F_1 as a new shape $\mathbf{S}_2^*$.
4: Calculate the final affine transformation parameter θ_g by projecting $\mathbf{S}_2^*$ to $\mathbf{S}^*$.

The proposed loss function $\mathcal{L}_{\theta_g}$ provides the original learning force for the convergence of $\hat{\theta}_g$. When $\hat{\theta}_g$ is close to the target θ_g^* after several training iterations, $\mathcal{L}_{\theta_g}$ is ignored, and the network keeps updating $\hat{\theta}_g$ only guided by the propagated errors of the subsequent layers. This strategy enhances the whole model by leveraging the merits of end-to-end learning. Examples of the normalized states are shown in Fig. 4. Compared with the original face images, the normalized states with frontal in-plane viewpoints and less unnecessary backgrounds are more similar to the frontal faces bounded by the ground truth shapes. This observation can be further confirmed by the mean face comparison between the original faces and the global reinitialization results, which are illustrated in Fig. 5 (a) and Fig. 5 (b), respectively. The reinitialization sub-network tends to generate well-normalized faces with similar appearances, apparent by its sharp mean face. In contrast, the original input faces located by face detectors only have a blurry mean face.

Global Regression Sub-network After obtaining F_g, the regression sub-network aims to regress the coarse shape $\mathbf{S}_g$ on F_g and its backbone of the sub-network can be built on different networks, *e.g.*, MobileNetV2 [60], VGG-S [61] or the ResNet-50 [62] network. It comprises a series of learnable convolutional and fully-connected layers. The output dimension of the last fully-connected layer is set to $2n$. Instead

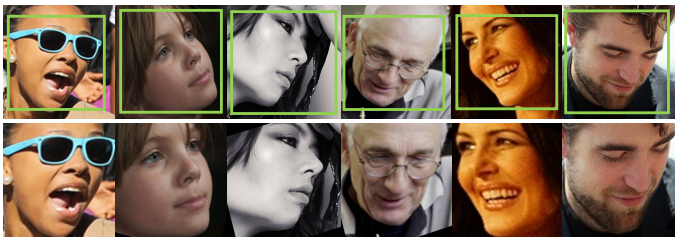


Fig. 4. Results of the global reinitialization sub-network. Top row: the input initial face images with the original face boxes. Bottom row: the transformed face images output by the global reinitialization sub-network.

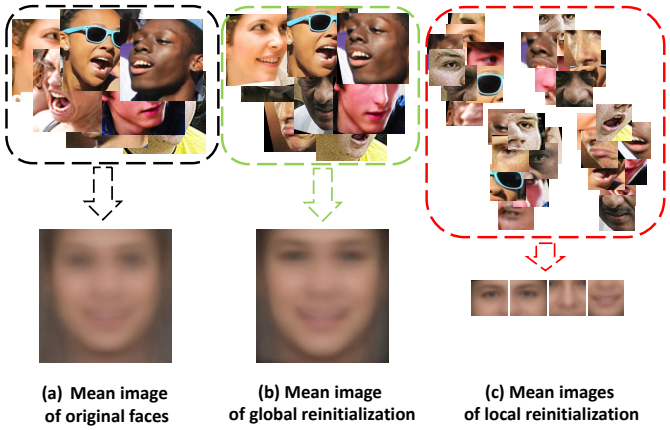


Fig. 5. Mean images of the original faces, the global and local reinitialization results.

of the standard ℓ_1 distance, shape increment normalized by the inter-ocular distance for faster convergence [28], is employed as the shape regression objective:

$$\mathcal{L}_g = \frac{||\Delta\hat{\mathbf{S}}_g - \Delta\mathbf{S}_g^*||_1}{d}, \tag{5}$$

where d is the inter-ocular distance of $\mathbf{S}_g^*$, $\Delta\hat{\mathbf{S}}_g = \hat{\mathbf{S}}_g - \mathbf{S}_{g_0}$, $\Delta\mathbf{S}_g^* = \hat{\mathbf{S}}_g^* - \mathbf{S}_{g_0}$. The variable $\mathbf{S}_{g_0}$, which is a mean shape calculated from the training dataset and transformed by $T_{\theta_g}^{-1}$, represents the initial shape at the global stage.

3.3 Local Shape Regression (LSR)

Face patches, *e.g.*, eyes, mouths, and noses, have notable appearance variations because of different identities, poses, and expressions. Although the GSR stage can predict fairly face shapes depending on the global reinitializations, it is not good at capturing the variations of local face patches. We introduce the LSR stage to reinitialize the local face patches to their normalized states and then regress for more accurate landmark positions.

All face landmarks of $\mathbf{S}_g$ are divided into five local shapes, *e.g.*, the shapes of the left eye $\mathbf{S}_g^1$, the right eye $\mathbf{S}_g^2$, the nose $\mathbf{S}_g^3$, and the mouth $\mathbf{S}_g^4$, and the contour $\mathbf{S}_g^5$. Note that the last face part does not go through the local reinitialization and regression sub-networks since it almost covers the whole face region, and the GSR stage has already learned its positions and shape well. We plot the definitions of the above four local patches on faces with different landmarks in Fig. 6 (a) and Fig. 6 (b).

Local Reinitialization Sub-network In this sub-network, different image patches with local shapes are independently

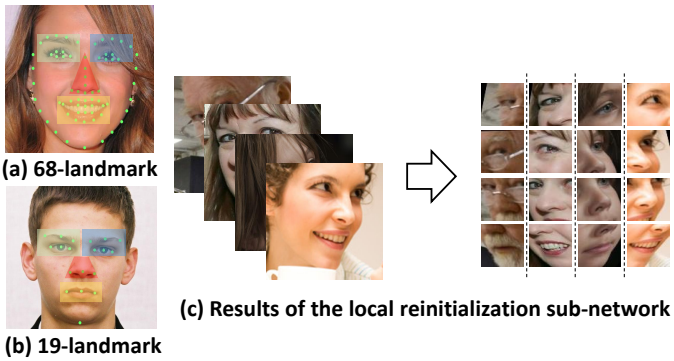


Fig. 6. Definitions of the four local patches on faces with different landmarks and output examples of the local reinitialization sub-network. The filled patterns with different colors on (a) and (b) represent different local patches.

reinitialized to their normalized states. The localization network in each branch of the sub-network is built on a simple structure consisting of three fully connected layers. Its input is the j^{th} local shape $\hat{\mathbf{S}}_g^j$, and the output is the local transformation matrix $T_{\theta_l^j}$. The local normalized state F_l^j is then produced by $T_{\theta_l^j}$ and the global state F_g. As the normalized results for the local patches indicated in Fig. 6 (c), they are all aligned to the consistent states with the similar centers of the local patches, frontal views, and small parts of the background textures. Compared with previous global reinitialization states shown in Fig. 5 (b), the current sub-network focuses on the local transformation for each face patch, as shown by the more clearly mean face patches in Fig. 5 (c). Therefore, it can boost the following regression sub-network ability to refine the landmark prediction on patch-level.

Local Regression Sub-network After obtaining F_l^j, the local regression sub-network learns to refine $\hat{\mathbf{S}}_g^j$ by minimizing the following loss function:

$$\mathcal{L}_l = ||\Delta\hat{\mathbf{S}}_l^j - \Delta\mathbf{S}_l^{j*}||_1, \tag{6}$$

where $\Delta\hat{\mathbf{S}}_l^j = \hat{\mathbf{S}}_l^j - \mathbf{S}_{l_0}^j$, $\Delta\mathbf{S}_l^{j*} = \mathbf{S}_l^{j*} - \mathbf{S}_{l_0}^j$. $\mathbf{S}_{l_0}^j$ represents the initial shape at the local stage, and it is transformed by $\hat{\mathbf{S}}_g^j$ and $T_{\theta_l^j}^{-1}$. A new layer, called a *Shape Inverse Transformer* layer, is introduced to obtain the final face shape $\hat{\mathbf{S}}^j$ from the regression result of LSR to permit end-to-end training. In this layer, $\hat{\mathbf{S}}_l^j$ on F_l^j is projected into the coordinate space of I by using Eqn. 7,

$$\hat{\mathbf{S}}^j = T_R T_{\theta_g} T_{\theta_l^j} \begin{pmatrix} \hat{\mathbf{S}}_l^j \\ 1 \end{pmatrix}, \tag{7}$$

where T_R is the rectangle geometric transformation that projects the points on F_0 into the coordinate space of I. Contour landmarks can be projected on I from F_g by using Eqn. (7) with $T_{\theta_l^j}$ omitted.

3.4 Adaptive Weighted Loss for Error-driven Learning

In this section, we first analyze the adverse effects on prediction accuracy brought by annotation inconsistencies and then introduce the proposed adaptive landmark-weighted loss.

Annotation Inconsistency Different facial landmarks have different annotation difficulties. The landmarks located at corners on the face region, *e.g.*, eye corners, and mouth corners, are relatively easy to be annotated. Landmarks located on the face contours or other plain areas are harder to be labeled because they have more degrees of freedom. The human annotators thus have larger inconsistencies for the positions of these landmarks.

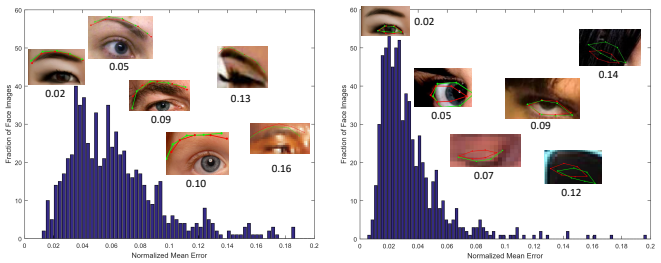
TABLE 1
Comparisons of alignment performance on different face parts using the NME metric. Results are obtained from Small MobileNetV2 trained on the 300-W training dataset using ℓ_1 loss.

δ \ Part	Contour	Eyebrows	Eyes	Nose	Mouth	All
0	8.65	6.44	3.47	4.25	4.35	5.57
3	8.65 ↑ 0.00%	6.47 ↑ 0.47%	3.53 ↑ 1.73%	4.35 ↑ **2.35**%	4.43 ↑ 1.84%	5.62 ↑ 0.90%
5	8.74 ↑ 1.04%	6.49 ↑ 0.78%	3.55 ↑ **2.31**%	4.30 ↑ 1.18%	4.40 ↑ 1.15%	5.63 ↑ 1.08%
10	8.99 ↑ 3.93%	6.73 ↑ 4.50%	4.02 ↑ **15.85**%	4.58 ↑ 7.76%	4.91 ↑ 12.87%	6.00 ↑ 7.72%

It is almost impossible to get the "ideal" annotations in practice. To analyze the impact of annotation inconsistency on alignment evaluation, we introduce other simulate inconsistencies to the existing dataset. The original landmark annotations on the 300-W training dataset are perturbed with different displacements in random directions within a radius range δ. All the face images are resized to a fixed size of 256×256 pixels before landmark perturbation. Then based on these "polluted" datasets with different perturbation magnitudes, we train several regression models using Small MobileNetV2 (see Section 4.1) with the ℓ_1 loss, to evaluate the alignment results on different face parts. Table 1 shows the comparison results on the 300-W full set measured by the metric of inter-pupil normalized mean errors (NME). The NME growth rates of different face parts caused by the additional perturbation vary with the increasing values of δ. Under the same perturbation magnitude, the prediction accuracy on eyes and mouth landmarks is more sensitive to label noises than on contour and eyebrows. It confirms our above observation of annotation inconsistency. Especially, contour landmarks only have a much less NME growth rate than other landmarks, revealing that the landmarks with more ambiguous annotations have a higher learning tolerance in model training.

The distribution of landmark prediction errors has a long tail phenomenon (see Fig. 7). The examples with large errors are only a small part of the overall dataset, but they significantly influence the regression learning and the final evaluation. Intuitively, compared to the examples with lower errors, the object learning should endow these "poorly performing" face images with larger weights to improve their prediction accuracy preferentially. According to the visualization results with NMEs larger than 0.05 in Fig. 7, large errors are often caused by ambiguous annotation instead of real inaccurate predictions. This situation is more prevalent in the landmarks which are not easily annotated (Fig. 7 (a)) against that in the landmarks with more precise definitions (Fig. 7 (b)).

Adaptive Weighted Loss The discouraging gradients from ambiguous annotations have a remarkable impact on the training samples with accurate landmark annotations. With the commonly used ℓ_1 and ℓ_2 loss functions, the regression



(a) Eyebrow Landmarks (b) Eye Landmarks

Fig. 7. The long tail phenomenon of prediction errors of different landmarks on 300-W. The green and red dots represent the ground truth and predicted landmarks on face images, respectively.

learning of high confident annotations is easily suppressed. To solve this problem, we propose an adaptive landmark-weighted loss function to make the network pay more attention to the stable landmarks and mitigate the inconsistent annotations' impacts. Formally, we introduce a weight term $\mathbf{w}$ to the original ℓ_1 loss function:

$$\mathcal{L} = \frac{||\mathbf{w} \cdot (\hat{\Delta\mathbf{S}} - \Delta\mathbf{S}^*)||_1}{d}, \tag{8}$$

where $\mathbf{w} = (w_{x_1}, \ldots, w_{x_n}; w_{y_1}, \ldots, w_{y_n})$ controls the constraint on each landmark according to its label reliability, which changes adaptively as the prediction error changes during the training procedure. In the following discussions, a landmark coordinate error is denoted as z for simplicity because each element w_z in $\mathbf{w}$ is defined on each coordinate independently.

In each training iteration, the statistics of landmark errors on a mini-batch are introduced to update w_z,

$$w_z = e^{-\sigma|z-\mu|}, \tag{9}$$

where μ and σ represent the mean and variance of landmark errors on M samples in the mini-batch predicted by the current training model. μ can be calculated as $\mu = \sum_{i=1}^{M} z^{(i)}/M$, while σ is calculate as $\sigma = \sqrt{\sum_{i=1}^{M}(z^{(i)} - \mu)^2/M}$. Note that w_z is inversely proportional to μ and σ. It can also be viewed as an approximated confidence score of the corresponding ground truth label estimated by the current iteration model.

The proposed loss function has three particular advantages: 1) The introduced weight term corresponding with annotation consistency controls the model's attention among landmarks. It partially avoids over-fitting on the landmark annotation with low confidence by defining a less constrained learning target. When all the elements in $\mathbf{w}$ equal to 1, the proposed loss becomes identical to the original ℓ_1 loss. 2) Without depending on any hyper-parameters manually set by trial and error, the prediction statistics of training checkpoints are defined to estimate the relative degree of annotation inconsistency in the training process and formulate an adaptive weight for each landmark. 3) Because of its simple form, the weighted loss can be easily used as a drop-in replacement of the standard ℓ_1 loss, *e.g.*, Eqn. (5) and Eqn. (6). The efficiency of our proposed novel loss is shown in Fig. 8. It noted that compared with the results predicted by the models trained with ℓ_1 and ℓ_2 loss functions, the proposed loss makes the predicted landmarks more compact around the real annotations.

4 Experiments

We implement the proposed deep regression architecture and conduct extensive evaluations to verify its effectiveness. In the following, we first introduce the experimental settings, including the benchmark datasets, evaluation metrics, and implementation details. Then we conduct a set of ablation studies to analyze the advantages of progressive reinitialization and error-driven learning. The generalization ability of the proposed architecture to different CNN backbones is also verified. Finally, we compare our method with other state-of-the-art methods on four popular benchmarks.

4.1 Experimental Settings

Benchmark Datasets Four public benchmark datasets, 300-W [63], AFLW [64], COFW [16], and WFLW [36], are adopted for evaluation in the experiments:

- *The 300W dataset* combines five existing datasets, iBug, LFPW, AFW, HELEN, and XM2VTS, and re-annotates them with 68 landmarks. Following the setting in [21], we use 3,148 images for training and 689 images for testing. The testing dataset is split into three parts: the common subset (554 images), the challenging subset (135 images), and the full set (689 images). The 300-W Challenge test set, which contains another 600 indoor and outdoor face images, is also used for further comparison.
- *The AFLW dataset* contains 24,386 face images with large variations in appearance (*e.g.*, pose, expression, ethnicity, and age) and environmental conditions. At most 21 landmarks are annotated for each face in the dataset. We ignore two landmarks on the ears and train our models with the remaining 19 landmarks. Following the experimental settings in [22], 20,000 images are used for training, while 4,386 images (AFLW-Full) and 1,314 images (AFLW-Frontal) are used for evaluation.
- *The COFW dataset* consists of face images with heavy occlusions and large shape variations. It is designed to evaluate face alignment in realistic conditions. We use the test set [65], which has 507 faces re-annotated with 68 landmarks as in 300-W, to further evaluate our model trained on the 300-W training set.
- *The WFLW dataset* are annotated with 98 landmarks with significant variations in expression, pose, and occlusion. It contains 10,000 faces, among which 7,500 faces are used for our model training and 2,500 faces for testing.

Evaluation Metrics To make fair comparisons with other face alignment methods, we adopt various evaluation metrics in the experiments, including the Normalized Mean Error (NME) [66], the Cumulative Errors Distribution (CED) curve [66], the Area Under the Curve (AUC) [29], and the Failure Rate (FR). The normalized error ϵ between the ground truth shape $\mathbf{S}^* = (\mathbf{p}_1^*, ..., \mathbf{p}_n^*)$ and the predicted shape $\hat{\mathbf{S}} = (\hat{\mathbf{p}}_1, ..., \hat{\mathbf{p}}_n)$ is defined as $\epsilon = \frac{1}{n}\sum_{i=1}^{n} ||\hat{\mathbf{p}}_i - \mathbf{p}_i^*||_2 / d_{\text{norm}}$,

where d_{norm} is the normalization factor. The NME of N samples can be averaged by all their normalized errors.

The CED curve is plotted by a cumulative distribution function $f(\epsilon)$ of the normalized error. The AUC metric is defined as $\text{AUC}_\alpha = \int_0^\alpha f(\epsilon)d\epsilon$, where α is the upper bound. The FR_α is defined as the fraction of the samples with normalized errors larger than α.

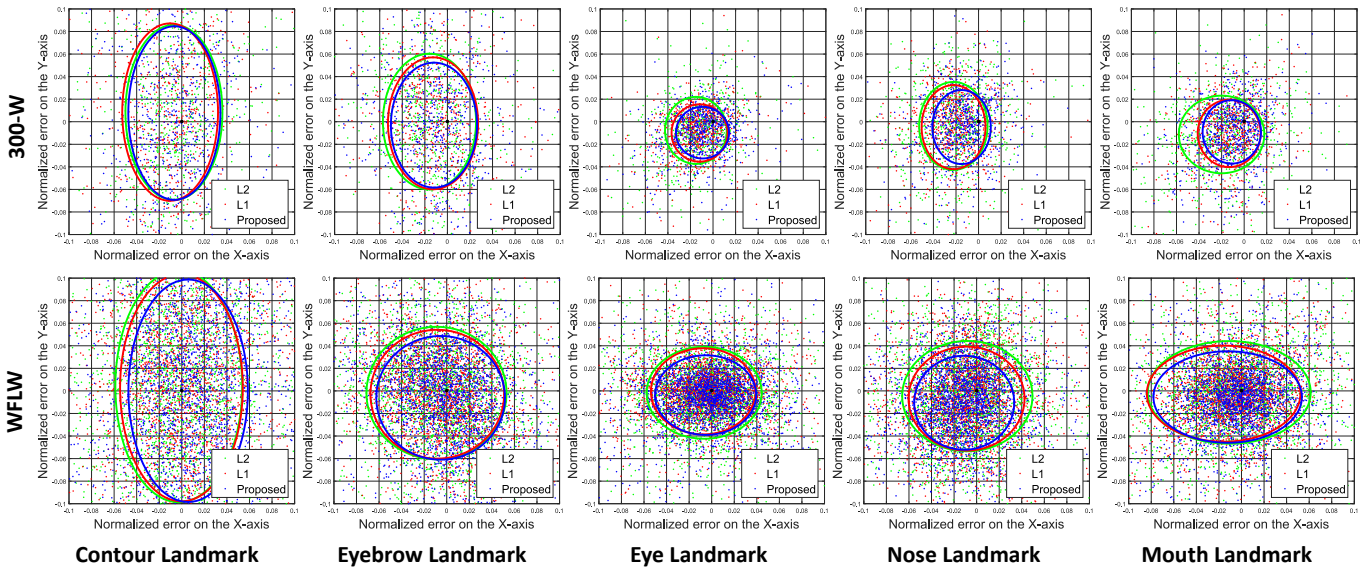


Fig. 8. Distributions of landmark-wise errors predicted by Small MobileNetV2 using different loss functions on 300-W and WFLW. The origin of coordinates (black dot) represents the ground truth. Each dot denotes a landmark offset. For each plotted ellipse, its center, semi-major axes, and semi-minor axes denote the mean value, the X-axis, and the Y-axis standard deviation of all corresponding landmarks errors, respectively. The landmarks from left to right columns are selected from the subsets {1, 18, 37, 31, 49} and {1, 34, 61, 55, 77} of the full annotations of 300-W (68-landmark) and WFLW (98-landmark), respectively.

On the 300-W and AFLW datasets, the NME and CED curve are used for evaluation. On the 300-W dataset, the inter-pupil distance is employed to normalize mean errors. On the AFLW dataset, as there are many profile faces in which the inter-pupil distance is approximating to zero, we follow the protocol in [22], use the face size instead as the normalizing factor. On the other datasets, including the 300-W test set, COFW, and WFLW, the NME metric is normalized by the inter-ocular distance [36]. AUC and FR with the default $\alpha = 0.1$ are also reported for comprehensive comparisons additionally.

Parameter Settings After detecting the faces in the input images, we crop the face regions from the face bounding boxes. To include more contextual information for alignment learning, we enlarge the face bounding box to a certain scale ratio of 1.2. Following the way of data augmentation in [1], we generate multiple samples for each training image by randomly perturbing the face boxes by translation and scaling. The distributions of perturbation amplitudes on the face boxes are simulated by the differences between the original boxes and the boxes bounded by the ground truth shapes. Besides, the training samples are also augmented by in-plane rotating the face images and landmarks simultaneously in the range [0°, 10°] randomly. By the operation of translation, scaling and rotation, 20 perturbed samples are generated for each training image. As points of the sampling grids in the reinitialization networks are normalized to [-1, 1] by face sizes, the predicted and ground truth shapes in the architecture are also transformed to the same coordinate space. A mini-batch size of 64, weight decay of 0.0002, momentum of 0.9, and epochs of 150 are adopted for model training.

Network Configurations. For the global reinitialization sub-network, the input is an image with $128 \times 128 \times 3$ pixels, and the output is a $256 \times 256 \times 3$ transformed face. The backbone of the global regression sub-network is called Small MobileNetV2, built on a simplified revision of MobileNetV2. To reduce the model size, Small MobileNetV2 resets the channels/strides of linear bottlenecks in the original MobileNetV2. The 7×7 average pooling layer in MobileNetV2 is further replaced with two fully-connected layers to enhance regression accuracy on the simplified network. The detailed configuration of the network is summarized in Table 2. This backbone is selected to emphasize the contributions of our reinitialization module and loss function as well as to allow direct comparison with other approaches. Because there are multiple branches in the local regression sub-networks for different facial parts, we explore a lightweight backbone consisting of six residual blocks and two fully connected layers to predict positions of local landmarks from image patches of $64 \times 64 \times 3$ pixels.

TABLE 2
Small MobileNetV2: Each line describes a sequence of identical layers, repeating q times. All layers in the same sequence have the same number c of output channels. The first layer of each sequence has a stride s and all others use stride 1. The expansion factor t is applied to all the inputs with different sizes.

Input	Operator	t	c	q	s
$256^2 \times 3$	conv2d 3x3	-	8	1	2
$128^2 \times 32$	bottleneck	1	8	1	1
$128^2 \times 8$	bottleneck	6	12	2	2
$64^2 \times 12$	bottleneck	6	16	2	2
$32^2 \times 16$	bottleneck	6	24	3	2
$16^2 \times 24$	bottleneck	6	32	3	2
$8^2 \times 32$	bottleneck	6	48	3	2
$4^2 \times 48$	bottleneck	6	64	2	2
$2^2 \times 48$	bottleneck	6	80	1	1
$2^2 \times 80$	conv2d 1x1	-	64	1	1
$2^2 \times 64$	fc	-	256	1	-
256	fc	-	256	1	-
256	fc	-	$2n$	1	-

Training Pipeline. There are four steps for training the whole architecture. The learning rate starts from 0.01 at the first three steps and 0.001 the last step, while a polynomial decay is adopted for dynamically adjusting the learning rate. ***In the first step***, the global reinitialization sub-network

TABLE 3
Ablation study of four different baseline models using four kinds of face detectors on the 300-W dataset using NME.

Detectors & Models	Common Subset	Challenging Subset	Full Set
MT_{B_1}	5.41	9.92	6.29
RF_{B_1}	5.52	9.57	6.31
OD_{B_1}	4.97	8.76	5.72
GT_{B_1}	4.83	8.58	5.56
MT_{B_2}	4.98	**8.81**	5.73
RF_{B_2}	4.99	**8.44**	5.67
OD_{B_2}	4.87	8.75	5.63
GT_{B_2}	4.85	8.71	5.60
MT_{GSR}	**4.86**	9.11	**5.70**
RF_{GSR}	**4.84**	8.78	**5.61**
OD_{GSR}	**4.86**	**8.21**	**5.52**
GT_{GSR}	**4.82**	**8.21**	**5.49**

is trained using PReLU [67] as the activation functions. ***In the second step***, the weights of the global reinitialization sub-network are fixed to train the global regression sub-network. Its weights are initialized with an ImageNet-pre-trained model. ***In the third step***, the weights of the GSR is fixed to train the LSR. Each reinitialization sub-network is supervised by the transformation parameters calculated from the pre-defined normalized face patches, while each local regression sub-network is trained from scratch. ***At the last step***, the whole network is fine-tuned from end to end by removing the loss layers in both the global and local reinitialization sub-networks.

4.2 Ablation Studies

4.2.1 Effectiveness of progressive reinitialization

To verify the advantages of the proposed reinitialization in the GSR stage, we train three different models on the 300-W dataset for comparison: a single shape regression network based on Small MobileNetV2 (denoted as B_1), a manually two-stage cascaded Small MobileNetV2 network (denoted as B_2), and the regression network using the global reinitialization sub-network (denoted as GSR). These models are learned with the ℓ_1 loss function, four types of face bounding boxes are used for evaluating model robustness : 1) ground truth bounding boxes (denoted as *GT*), which are the tight bounding boxes of the face shapes; 2) face boxes detected by the 300-W Official Detector (denoted as *OD*); 3) face boxes detected by the MTCNN detector [44] (denoted as *MT*); and 4) face boxes detected by the RetinaFace detector [68] (denoted as *RF*). The 300-W dataset itself provides the first two types of detection. The detectors of *MT* and *RF* have a few miss-detections. Their corresponding *OD* face boxes complement a portion of these miss-detections.

Table 3 shows the comparison results using the NME metric of the above three models with different face detectors. According to the table, the GSR model outperforms the baselines B_1 and B_2 by a wide margin in most cases. It shows that the GSR stage can provide better-normalized and more stable states for further shape regression than the original face boxes and hand-crafted reinitializations. It is also noted that the *GT* detector provides the best results for all three models, highlighting the importance of a good initialization for face shape regression.

To further verify the robustness of these models, we produce pseudo face boxes by perturbing the official detectors with different scales, translations, and rotations. We extend the face bounding boxes by a set of ratios, which ranges from 0.1 to 0.5. The results are compared in Table 4 (a). Then we apply a set of random ratios from 0.05 to 0.25 of face box size to translate face boxes, and the results are compared in Table 4 (b). We also rotate face images in-plane from 0° to 25° to evaluate these methods under various in-plane rotations, with the results compared in Table 4 (c). Among these methods, the proposed GSR model exhibits the best robustness to various inputs with different spatial transformations. Especially under extreme poor input with the scale ratio of 0.5 or the translation ratio of 0.25, GSR still achieves the accuracy with about 43% and 20% improvements, respectively, over the second-best baseline B_2.

The above two sets of experimental results demonstrate the robust performance of our GSR stage; the effectiveness of the LSR stage also needs to be evaluated. Table 5 shows the comparison among the above baselines and the regression model with two-stage progressive reinitialization on the 300-W and WFLW datasets. $G\&LSR$ denotes the combination of the global and local sub-networks, and $G\&LSR^*$ denotes $G\&LSR$ without using the local reinitialization sub-network.

According to the table, it is noted that the $G\&LSR$ model further obtains respectively 4.2% and 3.0% improvements over GSR on the 300-W and WFLW datasets. This result shows our LSR stage can further improve face alignment accuracy by the reinitialization and finer regression on patch-level. Especially, without the local reinitialization sub-network, $G\&LSR^*$ has only 2.4% and 1.2% improvements over GSR on both datasets and even has an accuracy drop on the 300-W challenging subset. The comparison verifies the importance of the local reinitialization in the LSR stage. Some visual face alignment examples with large variations in face view, expression, illumination, and occlusion, which are predicted by $G\&LSR$, GSR and the baseline method B_1 are shown in Fig. 9.

4.2.2 Robustness to annotation inconsistency

In this ablation study, we verify the effectiveness of the proposed adaptive weighted loss function on alleviating the annotation inconsistency problem in different training datasets. On the 300-W and WFLW training datasets, three baseline models are trained for comparison. They are all based on the backbone network of Small MobileNetV2 but are learned with the ℓ_2, ℓ_1, and the adaptive weighted loss function, respectively. ℓ_1 helps the model obtain better performance than that of ℓ_2 on the both datasets, while the adaptive loss function further improves ℓ_1 with about 6.7% and 6.0% reductions in NME of the 300-W (see Fig. 10 (a)) and WFLW datasets (see Fig. 10 (b)).

It is reasonable that the proposed loss function improves the regression accuracy of landmarks on eyes and mouths, which have clearer annotation definitions and get more prediction penalties than that of the other landmarks in training procedures. The landmarks on face contours and eyebrows, whose regression learning tasks are distributed with smaller weights tend to be easier in the training procedure, also have slight declines in prediction errors (see Fig. 10 (c) and Fig. 10 (d)).These comparisons verify that the error-driven learning, which treats different landmarks with adjustable

TABLE 4
Comparison of NME(%) on 300-W based on different kinds of face box perturbations.

(a) Different Scales

Scale	0.1	0.2	0.3	0.4	0.5
B_1	5.75	6.36	8.15	10.93	15.72
B_2	5.64	5.65	6.18	8.29	11.80
GSR	**5.51**	**5.60**	**5.74**	**6.03**	**6.67**

(b) Different Translations

Translation	0.05	0.10	0.15	0.20	0.25
B_1	5.56	5.64	6.07	7.92	12.34
B_2	5.62	5.62	5.72	5.90	8.11
GSR	**5.48**	**5.52**	**5.62**	**5.84**	**6.52**

(c) Different Rotations

Rotation (°)	5	10	15	20	25
B_1	5.62	5.72	5.88	5.92	6.01
B_2	5.57	5.60	5.82	5.92	6.00
GSR	**5.51**	**5.52**	**5.53**	**5.58**	**5.60**

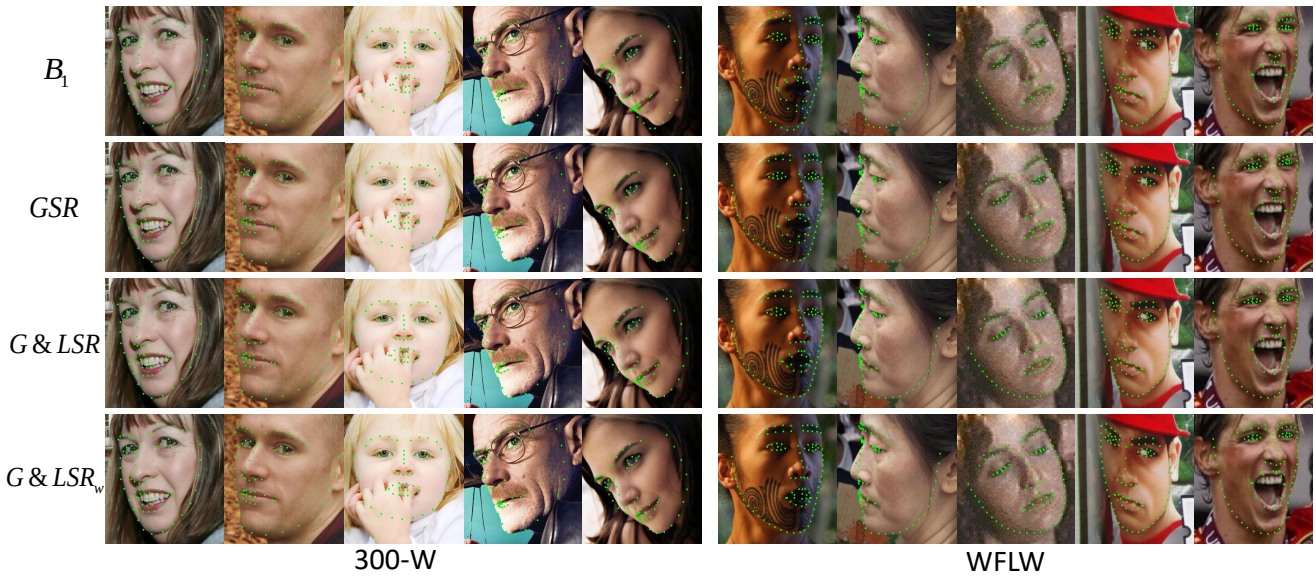


Fig. 9. Visual comparison of face alignment results predicted by different models on 300-W and WFLW.

TABLE 5
Comparison of our proposed $G\&LSR$ model and other baselines on 300-W and WFLW using NME(%).

Models	300-W			WFLW
	Common	Challenging	Full	
B_1	4.97	8.76	5.72	6.51
GSR	4.86	8.21	5.52	5.69
$G\&LSR^*$	4.67	8.33	5.39	5.62
$G\&LSR$	**4.57**	**8.21**	**5.29**	**5.52**

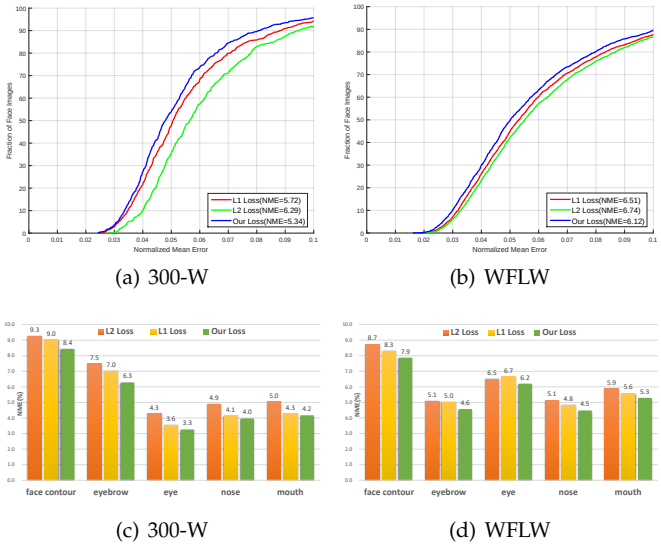
constraints according with their label reliabilities, makes full use of strong learning ability of deep convolutional networks, and reduces over-fitting caused by annotation inconsistencies in training datasets.

Since the model with the adaptive weighted loss function obtains the best performance compared with the other models, we use it as the final regression loss to train the above $G\&LSR$ model. We use $G\&LSR_\omega$ to denote the combination of the progressive reinitialization and the adaptive weighted loss in the following sections. Compared with the results of $G\&LSR$, $G\&LSR_\omega$ achieves the NMEs (%) of 5.17 and 5.26 on the 300-W and WFLW datasets, respectively, significant outperforming the results (5.29 and 5.52) predicted by $G\&LSR$. The last row in Fig. 9 shows the qualitative results detected by $G\&LSR_\omega$.

4.2.3 Generalization to different backbone networks

In this ablation study, we evaluate the generalization ability of the proposed deep regression architecture to different backbone networks. The analysis also helps to find a better trade-off between the model accuracy and efficiency, and makes the following comparisons with other state-of-the-art methods comprehensively. We replace the regression backbone in $G\&LSR_\omega$ with a medium-size MobileNetV2 and a large-size ResNet-50 [62] to build another two variants of the proposed method, $G\&LSR_\omega$/M, and $G\&LSR_\omega$/L. For $G\&LSR_\omega$/M, we retain most configurations of original MobileNetV2 while resetting the stride of the sixth bottleneck to 2 and replacing the 7×7 average pooling layer with two fully-connected layers. Considering the strong learning ability, we do not apply the LSR stage for the ResNet-50 backbone, and still call it $G\&LSR_\omega$/L for convenience.

Table 6 depicts the accuracy enhancements by our proposed progressive module and error-driven loss function



(a) 300-W (b) WFLW

(c) 300-W (d) WFLW

Fig. 10. CED curves and NME (%) comparison of different landmark types predicted by the Small MobileNetV2 backbone learned with different loss functions on 300-W and WFLW.

TABLE 6
Comparison of the proposed method based on different backbones on 300-W.

Backbone	Progressive Reinitialization	Weighted Learning	NME (%)
Small	✗	✗	5.72
	✔	✗	5.29
	✔	✔	**5.17**
Medium	✗	✗	5.02
	✔	✗	4.78
	✔	✔	**4.61**
Large	✗	✗	4.35
	✔	✗	4.15
	✔	✔	**3.95**

for the above backbones of different sizes. According to the table, our methods outperform the original architectures in accuracy for all the networks. Among these models, the progressive reinitialization gets the largest 7.52% improvement for Small MobileNetV2, while obtaining NMEs reductions of 4.78% and 4.60% respectively for MobileNetV2 and ResNet-50. The results demonstrate that the reinitialization module is useful to unstable initializations for models of different sizes. The weighted loss function gets the 4.82% additional improvement for the large capability ResNet-50 network, while there are only 2.27% and 3.56% improvements for the small-size and medium-size backbones, respectively. It shows that applying the weighted loss to the large network can be more beneficial to reduce the impacts of annotation inconsistencies in training.

We further evaluate the effectiveness of the models mentioned above on an Nvidia Tesla P100 GPU, an Intel Core i7 CPU@2.20 GHz and an iPhone 6S mobile device. The GPU evaluation is based on the Matlab interface of Caffe [69], and the CPU and mobile evaluation is based on MNN [70]. Table 7 shows the results in different model sizes and speeds. We can see that even though our method combines a group of modules, $G\&LSR_\omega$ has only a 5.9 MB model size and still achieves a very high speed of 98.0 fps on the mobile device. Meanwhile, $G\&LSR_\omega$/M with a 15.7 MB model size costs about 4.5 ms (over 222 fps) on the mobile device, is also suitable for mobile applications with higher accuracy requirements. With a 96.7 MB model size, $G\&LSR_\omega$/L runs at the speed of 80.6 fps (GPU) and 15.6 fps (CPU). On GPU, the speed advantage of our method seems not obvious. On CPU and mobile devices, our method with the well-designed architectures shows its superiority over other recent algorithms, including our original conference paper TSR [32]. Even the large-size model $G\&LSR_\omega$/L, cost only about 208 ms on the mobile, faster than the other methods with backbones with equal or larger model sizes. These results of the proposed deep regression architecture not only verify its generalization ability but also show its flexibility to different application scenarios.

4.3 Comparisons with the State-of-the-arts

We now compare the proposed deep architecture with other existing state-of-the-art methods on the 300-W, 300-W test, AFLW, COFW, and WFLW datasets. The results cited in the following benchmarks for comparison are all reported to use the same experimental setting and evaluation protocol like ours. The results of different models based on our approach are reported for a comprehensive evaluation of the proposed deep architecture.

TABLE 7
Comparison of our proposed method based on different networks in terms of model size and speed. RWing* indicates a fast variant of RWing using a two-stage plain CNN framework.

Network	Model Parameters	FLOPs	Model Size (MB)	Speed(fps)		
				GPU	CPU	Mobile
SAN [34]	-	-	798.5	2.9	-	-
LAB [36]	-	-	50.7	16.7	3.9	-
HRNet [71]	9.3M	4.3G	-	-	-	
RWing [54]	32M	3.87G	122	154	12	0.62
RWing*	178M	8.04G	680	1010	10	0.36
TSR [32]	102.07M	2.74G	407.4	172	17.1	4.6
B_1	335.4K	48.5M	1.3	333.3	268.1	166.7
GSR	593.2K	52.6M	2.3	250.0	243.9	142.3
$G\&LSR_\omega$	1.83M	57.7M	5.9	149.3	180.5	98.0
$G\&LSR_\omega$/M	2.91M	0.27G	15.7	222.2	83.5	25.6
$G\&LSR_\omega$/L	24.39M	5.28G	96.7	80.6	15.6	4.8

Results on 300-W Table 8 shows the NME comparison results on the 300-W dataset. All of our models use the *OD* face boxes as initialization and achieve comparable performance with the same experimental setting with other methods. Due to the effective backbone and the novel loss function, the performance of $G\&LSR_\omega$ is slightly behind the preliminary version TSR, but runs ten times faster on CPU and twenty times faster on mobile devices. $G\&LSR_\omega$/M also outperforms TSR by a 7.7% accuracy improvement and costs only about a fifth of the running time. By using a backbone with the similar complexity with WING [35], our large-size model $G\&LSR_\omega$/L obtains the best performance on the full set than the other existing approaches, *e.g.*, AWing [55], WING [35], SBR [53], and SA [38]. The comparison demonstrates the strong generalization ability of our method to deal with face alignment under wild environments.

TABLE 8
Comparison on 300-W using NME (%).

Method	Common Subset	Challenging Subset	Full Set
DeepReg [72]	4.51	13.80	6.31
LBF [31]	4.95	11.98	6.32
CFSS [21]	4.73	9.98	5.76
TCDCN [45]	4.80	8.60	5.54
DDN [26]	-	-	5.59
MDM [73]	4.83	10.14	5.88
HSLE [40]	3.94	7.24	4.59
LPR [49]	3.83	7.46	4.54
SSST [39]	3.98	7.21	4.54
AWing [55]	3.77	6.52	4.31
LAB [36]	3.42	6.98	4.12
SBR [53]	3.28	7.10	4.10
WING [35]	**3.27**	7.18	4.04
SA [38]	3.45	**6.38**	4.02
TSR [32]	4.36	7.56	4.99
$G\&LSR_\omega$	4.52	7.82	5.17
$G\&LSR_\omega$/M	4.06	6.87	4.61
$G\&LSR_\omega$/L	3.34	6.40	**3.95**

Results on 300-W test and COFW The above-trained models are also evaluated on other 68-landmark datasets, 300-W test, and COFW. The AUC and Failure Rate metrics are used for the 300-W test dataset, and the results are reported in Table 9 (a). Among all the methods, our model $G\&LSR_\omega$/L

TABLE 9
Comparison on 300-W test and COFW using NME (%), AUC and Failure Rate (%). Symbol "+" indicates the model is trained with external data.

(a) 300-W test

Method	AUC	Failure Rate
Zhou *et al*. [48]	0.3281	13.00
Deng *et al*. [74]	0.4752	5.5
DenseReg [75]	0.5219	3.67
3FabRec [42]	0.5461	0.17
JMFA [47]	0.5485	1.00
LAB [36]	0.5885	0.83
RWing [54]	0.5923	0.50
JMFA$^+$	0.6071	0.33
$G\&LSR_\omega$	0.5504	0.67
$G\&LSR_\omega$/M	0.5895	0.17
$G\&LSR_\omega$/L	**0.6280**	**0.17**

(b) COFW

Method	NME	Failure Rate
RCPR [19]	8.76	20.12
TCDCN [45]	7.66	16.17
HPM [65]	6.72	6.71
CFSS [21]	6.28	9.07
LAB [36]	4.62	2.17
ODN [37]	5.30	-
SSST [39]	4.43	2.82
HRNet [71]	**3.45**	**0.19**
$G\&LSR_\omega$	4.50	0.98
$G\&LSR_\omega$/M	4.30	0.59
$G\&LSR_\omega$/L	4.21	0.20

achieves the best performance in terms of AUC and Failure Rate, and $G\&LSR_\omega$ also has an acceptable result. It is worth mentioning that the performance of $G\&LSR_\omega$/M is better than most of other recently proposed methods including LAB [36], while $G\&LSR_\omega$/M has a smaller model size than LAB (15.7 MB vs. 50.7 MB) and much faster processing speed (250 fps vs. 60 fps).

The NME and Failure Rate metrics for the COFW dataset are reported in Table 9 (b). According to the table, $G\&LSR_\omega$/L and $G\&LSR_\omega$/M are ranked second and third respectively on the evaluation list, only slightly behind HRNet [71]. The three models' outstanding performance shows the robustness of our approach against faces with different poses and various occlusions.

Results on AFLW The comparisons on AFLW are reported in Table 10 and Fig. 11. The three model variants based on our method improve a lot of our original conference paper. Compared to other state-of-the-art approaches, $G\&LSR_\omega$/L achieves the 13.2% and 7.1% improvements of the second-best RWing method under both the AFLW-Full and AFLW-Frontal settings, while $G\&LSR_\omega$/M and $G\&LSR_\omega$ also have competitive performances. $G\&LSR_\omega$/M can even be ranked the fourth in the table, following the recently proposed methods AWing and RWing. It is noted that the backbone of $G\&LSR_\omega$/M is a lightweight network, while AWing and RWing use large-capacity backbones, such as four stacks of HG or ResNet-50. As shown in Fig. 11, the smallest and fastest model $G\&LSR_\omega$ outperforms TSR by a large margin and is only slightly worse than the Wing method. The results show that our method enhances the regression robustness of sparse facial landmarks.

Results on WFLW Table 11 compares the results of different methods on the WFLW dataset. On this 98-landmark dataset, $G\&LSR_\omega$/L significantly outperforms all the competing approaches in terms of all three metrics, improving the NME of AWing respectively by 8.7%, the Failure Rate of AWing by 64.7%, and the AUC of SSST [39] by 2.1%. $G\&LSR_\omega$ with only a 5.9MB model performs better than the recent methods with larger models, *e.g.*, LAB (50.7 MB) and 3FabRec [42] (20 MB) in NME and Failure Rate. $G\&LSR_\omega$/M even obtains the fourth-best Failure Rate in the table, while the methods with higher rankings suffer from larger FLOPs (HRNet with 4.3G FLOPs vs. $G\&LSR_\omega$/M with 0.27G FLOPs), or lower running speed on GPU (AWing with 34.5 fps vs. $G\&LSR_\omega$/M with 222.2 fps). The comparisons of our model variants and other approaches indicates the effectiveness of the proposed structure.

TABLE 10
Comparison on AFLW using NME (%).

Method	AFLW-Full	AFLW-Frontal
CCL [22]	2.72	2.17
SAN [34]	1.91	1.85
SBR [53]	2.14	-
TS^3 [41]	1.99	1.86
LAB [36]	1.85	1.62
ODN [37]	1.63	1.38
SA [38]	1.60	-
3FabRec [42]	1.84	1.59
HRNet [71]	1.57	1.46
AWing [55]	1.53	1.38
RWing [54]	1.51	1.27
TSR [32]	2.17	-
$G\&LSR_\omega$	1.67	1.41
$G\&LSR_\omega$/M	1.55	1.32
$G\&LSR_\omega$/L	**1.31**	**1.18**

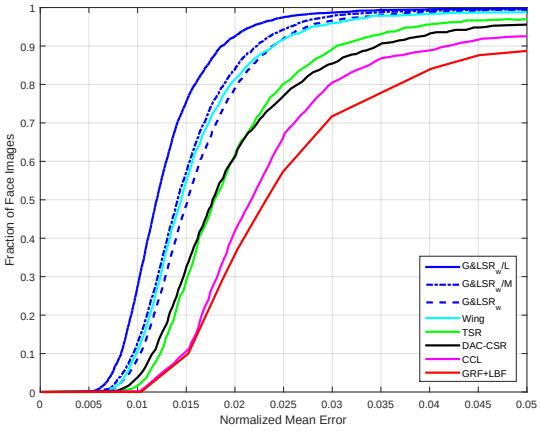


Fig. 11. CED curves of our method compared with other state-of-the-art methods on AFLW. These methods include Wing [35], TSR [32], DAC-CSR [76], CCL [22], and GRF+LBF [77].

TABLE 11
Comparisons on WFLW using NME, Failure Rate, and AUC.

Method	NME (%)	Failure Rate (%)	AUC (%)
DVLN [78]	6.08	10.84	0.4551
LAB [36]	5.27	7.56	0.5323
Wing [35]	5.11	6.00	0.5504
3FabRec [42]	5.62	8.28	0.484
SSST [39]	4.39	4.08	0.5913
RWing [54]	4.99	5.64	0.5585
DeCaFA [51]	4.62	4.84	0.563
HRNet [71]	4.60	-	-
LUVLi [56]	4.37	3.12	0.577
AWing [55]	4.36	2.84	0.5719
$G\&LSR_\omega$	5.26	5.72	0.4925
$G\&LSR_\omega$/M	4.84	3.92	0.5255
$G\&LSR_\omega$/L	**3.98**	**1.00**	**0.6042**

5 Conclusion

This paper focuses on face shape initialization and the learning of objective functions for face alignment. Previous work has not examined these two aspects in depth. We have presented a deep regression architecture consisting of two progressive reinitialization stages, which exhibit strong robustness to various face detection initialization. We propose an adaptive landmark-weighted loss function to obtain

error-sensitive embedding for improving face alignment training on a face dataset with unreliable annotations. The proposed method achieves state-of-the-art performance on several benchmarks of face alignment. It has good generalization ability and runs at real-time speed on mobile devices. We plan to improve the proposed deep regression architecture by introducing 3D transformation for faces with arbitrary poses and building a semi-supervised architecture for learning face alignment on training data that contains many unlabeled face images.

REFERENCES

[1] X. Xiong and F. Torre, "Supervised descent method and its applications to face alignment," in *CVPR*, 2013, pp. 532–539.

[2] S. Xiao, S. Yan, and A. A. Kassim, "Facial landmark detection via progressive initialization," in *ICCVW*, 2015, pp. 986–993.

[3] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: real-time face capture and reenactment of rgb videos," in *CVPR*, 2016, pp. 2387–2395.

[4] I. Korshunova, W. Shi, J. Dambre, and L. Theis, "Fast face-swap using convolutional neural networks," in *ICCV*, 2017, pp. 3677–3685.

[5] G. B. Huang and E. Learned-Miller, "Labeled faces in the wild: updates and new reporting procedures," Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep. UM-CS-2014-003, 2014.

[6] X. Zhu, Z. Lei, J. Yan, D. Yi, and S. Z. Li, "High-fidelity pose and expression normalization for face recognition in the wild," in *CVPR*, 2015, pp. 787–796.

[7] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models-their training and application," *CVIU*, vol. 61, no. 1, pp. 38–59, 1995.

[8] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," *TPAMI*, vol. 23, no. 6, pp. 681–685, 2001.

[9] D. Cristinacce and T. Cootes, "Automatic feature localization with constrained local models," *PL*, vol. 41, no. 10, pp. 3054–3067, 2008.

[10] P. Dollar, P. Welinder, and P. Perona, "Cascaded pose regression," *ICCV*, vol. 238, no. 6, pp. 1078–1085, 2010.

[11] J. Yan, Z. Lei, D. Yi, and S. Li, "Learn to combine multiple hypotheses for accurate face alignment," in *ICCVW*, 2013, pp. 392–396.

[12] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *CVPR*, 2012, pp. 2879–2886.

[13] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face alignment by explicit shape regression," *IJCV*, vol. 107, no. 2, pp. 177–190, 2014.

[14] J. Xing, Z. Niu, J. Huang, W. Hu, and S. Yan, "Towards multi-view and partially-occluded face alignment," in *ICCV*, 2014, pp. 1829–1836.

[15] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Learning deep representation for face alignment with auxiliary attributes," *TPAMI*, vol. 38, no. 5, pp. 918–930, 2016.

[16] X. P. Burgos-Artizzu, P. Perona, and P. Dollár, "Robust face landmark estimation under occlusion," in *ICCV*, 2013, pp. 1513–1520.

[17] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face alignment at 3000 fps via regressing local binary features," in *CVPR*, 2014, pp. 1685–1692.

[18] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *CVPR*, 2014, pp. 1867–1874.

[19] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Robust discriminative response map fitting with constrained local models," in *ICCV*, 2013, pp. 3444–3451.

[20] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment," in *ECCV*, 2014, pp. 1–16.

[21] S. Zhu, C. Li, C. C. Loy, and X. Tang, "Face alignment by coarse-to-fine shape searching," in *ICCV*, 2015, pp. 4998–5006.

[22] S. Zhu, C. Li, C.-C. Loy, and X. Tang, "Unconstrained face alignment via cascaded compositional learning," in *CVPR*, 2016, pp. 3409–3417.

[23] J. Xing, Z. Niu, J. Huang, W. Hu, X. Zhou, and S. Yan, "Towards robust and accurate multi-view and partially-occluded face alignment," *TPAMI*, vol. 40, no. 4, pp. 987–1001, 2018.

[24] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Incremental face alignment in the wild," in *CVPR*, 2014, pp. 1859–1866.

[25] H. Yang, C. Zou, and I. Patras, "Face sketch landmarks localization in the wild," *SPL*, vol. 21, no. 11, pp. 1321–1325, 2014.

[26] X. Yu, F. Zhou, and M. Chandraker, "Deep deformation network for object landmark localization," in *ECCV*, 2016, pp. 52–70.

[27] S. Xiao, J. Feng, J. Xing, H. Lai, S. Yan, and A. Kassim, "Robust facial landmark detection via recurrent attentive-refinement networks," in *ECCV*, 2016, pp. 57–72.

[28] Y. Wu, T. Hassner, K. Kim, G. Medioni, and P. Natarajan, "Facial landmark detection with tweaked convolutional neural networks," *TPAMI*, vol. 40, no. 12, pp. 3067–3074, 2018.

[29] H. Yang, X. Jia, C. C. Loy, and P. Robinson, "An empirical study of recent face alignment methods," *arXiv preprint arXiv:1511.05049*, 2015.

[30] H. Yang, W. Mou, Y. Zhang, I. Patras, H. Gunes, and P. Robinson, "Face alignment assisted by head pose estimation," in *BMVC*, 2015, pp. 130.1–130.13.

[31] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face alignment via regressing local binary features," *TIP*, vol. 25, no. 3, pp. 1233–1245, 2016.

[32] 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 *CVPR*, 2017, pp. 3691–3700.

[33] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *CVPR*, 2013, pp. 3476–3483.

[34] X. Dong, Y. Yan, W. Ouyang, and Y. Yang, "Style aggregated network for facial landmark detection," in *CVPR*, 2018, pp. 379–388.

[35] 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 *CVPR*, 2018, pp. 2235–2245.

[36] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou, "Look at boundary: a boundary-aware face alignment algorithm," in *CVPR*, 2018, pp. 2129–2138.

[37] M. Zhu, D. Shi, M. Zheng, and M. Sadiq, "Robust facial landmark detection via occlusion-adaptive deep networks," in *CVPR*, 2019, pp. 3486–3496.

[38] Z. Liu, X. Zhu, G. Hu, H. Guo, M. Tang, Z. Lei, N. M. Robertson, and J. Wang, "Semantic alignment: Finding semantically consistent ground-truth for facial landmark detection," in *CVPR*, 2019, pp. 3467–3476.

[39] S. Qian, K. Sun, W. Wu, C. Qian, and J. Jia, "Aggregation via separation: Boosting facial landmark detector with semi-supervised style translation," in *ICCV*, 2019, pp. 10 153–10 163.

[40] X. Zou, S. Zhong, L. Yan, X. Zhao, J. Zhou, and Y. Wu, "Learning robust facial landmark detection via hierarchical structured ensemble," in *ICCV*, 2019, pp. 141–150.

[41] X. Dong and Y. Yang, "Teacher supervises students how to learn from partially labeled images for facial landmark detection," in *ICCV*, 2019, pp. 783–792.

[42] B. Browatzki and C. Wallraven, "3fabrec: Fast few-shot face alignment by reconstruction." in *CVPR*, 2020.

[43] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun, "Joint cascade face detection and alignment," in *ECCV*, 2014, pp. 109–122.

[44] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *SPL*, vol. 23, no. 10, pp. 1499–1503, 2016.

[45] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *ECCV*, 2014, pp. 94–108.

[46] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *TPAMI*, vol. 41, no. 1, pp. 121–135, 2019.

[47] J. Deng, G. Trigeorgis, Y. Zhou, and S. Zafeiriou, "Joint multi-view face alignment in the wild," *TIP*, vol. 28, no. 7, pp. 3636–3648, 2019.

[48] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, "Extensive facial landmark localization with coarse-to-fine convolutional network cascade," in *ICCVW*, 2013, pp. 386–391.

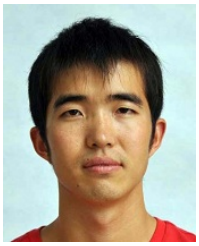
[49] Z. Huang, E. Zhou, and Z. Cao, "Coarse-to-fine face alignment with multi-scale local patch regression," *arXiv preprint arXiv:1511.04901*, 2015.

[50] M. Kowalski, J. Naruniec, and T. Trzcinski, "Deep alignment network: a convolutional neural network for robust face alignment," in *CVPRW*, 2017, pp. 88–97.

[51] A. Dapogny, M. Cord, and K. Bailly, "Decafa: Deep convolutional cascade for face alignment in the wild," in *ICCV*, 2019, pp. 6893–6901.

[52] L. Chen, H. Su, and Q. Ji, "Face alignment with kernel density deep neural network," in *ICCV*, 2019, pp. 6992–7002.

[53] 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 *CVPR*, 2018, pp. 360–368.
[54] Z. Feng, J. Kittler, M. Awais, and X. Wu, "Rectified wing loss for efficient and robust facial landmark localisation with convolutional neural networks," *IJCV*, pp. 1–20, 2019.
[55] X. Wang, L. Bo, and L. Fuxin, "Adaptive wing loss for robust face alignment via heatmap regression," in *ICCV*, 2019, pp. 6971–6981.
[56] A. Kumar, T. Marks, W. Mou, Y. Wang, M. Jones, A. Cherian, T. Koike-Akino, X. Liu, and C. Feng, "Luvli face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood," in *CVPR*, 2020.
[57] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *NeurIPS*, 2015, pp. 2017–2025.
[58] X. Yu and F. Porikli, "Face hallucination with tiny unaligned images by transformative discriminative neural networks," in *AAAI*, vol. 31, no. 1, 2017.
[59] X. Yu, F. Shiri, B. Ghanem, and F. Porikli, "Can we see more? joint frontalization and hallucination of unaligned tiny faces," in *TPAMI*, vol. 40, no. 4, 2019, pp. 2148–2164.
[60] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: inverted residuals and linear bottlenecks," in *CVPR*, 2018, pp. 4510–4520.
[61] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: delving deep into convolutional nets," *BMVC*, 2014.
[62] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.
[63] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: the first facial landmark localization challenge," in *ICCVW*, 2013, pp. 397–403.
[64] M. Koestinger, 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 *ICCVW*, 2011, pp. 2144–2151.
[65] G. Ghiasi and C. C. Fowlkes, "Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model," in *CVPR*, 2014, pp. 2385–2392.
[66] D. Cristinacce and T. Cootes, "Feature detection and tracking with constrained local models." in *BMVC*, vol. 1, no. 2, 2006, p. 3.
[67] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV*, 2015, pp. 1026–1034.
[68] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-shot multi-level face localisation in the wild," in *CVPR*, 2020.
[69] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.
[70] X. Jiang, H. Wang, Y. Chen, Z. Wu, L. Wang, B. Zou, Y. Yang, Z. Cui, Y. Cai, T. Yu, C. Lv, and Z. Wu, "Mnn: A universal and efficient inference engine," in *MLSys*, 2020.
[71] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang, "High-resolution representations for labeling pixels and regions," *arXiv: Computer Vision and Pattern Recognition*, 2019.
[72] B. Shi, X. Bai, W. Liu, and J. Wang, "Deep regression for face alignment," *TNNLS*, vol. 29, no. 1, pp. 183–194, 2018.
[73] 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 *CVPR*, 2016, pp. 4177–4187.
[74] J. Deng, Q. Liu, J. Yang, and D. Tao, "M3 CSR: multi-view, multi-scale and multi-component cascade shape regression," *IVC*, vol. 47, pp. 19–26, 2016.
[75] R. Alp Guler, G. Trigeorgis, E. Antonakos, P. Snape, S. Zafeiriou, and I. Kokkinos, "DenseReg: fully convolutional dense shape regression in-the-wild," in *CVPR*, 2017, pp. 6799–6808.
[76] Z.-H. Feng, J. Kittler, W. Christmas, P. Huber, and X.-J. Wu, "Dynamic attention-controlled cascaded shape regression exploiting training data augmentation and fuzzy-set sample weighting," in *CVPR*, 2017, pp. 2481–2490.
[77] K. Hara and R. Chellappa, "Growing regression forests by classification: applications to object pose estimation," in *ECCV*, 2014, pp. 552–567.
[78] W. Wu and S. Yang, "Leveraging intra and inter-dataset variations for robust face alignment," in *CVPRW*, 2017, pp. 150–159.



Xiaohu Shao received the B.E. degree in telecommunication engineering from China University of Geosciences in 2009, the M.E. degree in Signal and information processing from University of Electronic Science and Technology of China in 2012, and the Ph.D. degree in computer application technology from Chongqing Institute of Green and Intelligent Technology (CIGIT), Chinese Academy of Sciences in 2021. He is currently a research assistant at CIGIT, Chinese Academy of Sciences. His research interests include face alignment and recognition.

Junliang Xing received his dual B.S. degrees in computer science and mathematics from Xi'an Jiaotong University, Shaanxi, China, in 2007, and the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2012. He is currently a Professor with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests mainly focus on computer vision problems related to human faces and computer gaming problems in imperfect information decision.

Jiangjing Lyu is a senior algorithm engineer in Alibaba Group. He received the B.S. degree in Information and computing science from University of Science and Technology of Hunan, China, the Ph.D. degree in computer science from University of Chinese Academy of Sciences, China, in 2012 and 2017 respectively. His research interests include face recognition and 3D face reconstruction.

Xiangdong Zhou is an associate professor at CIGIT, Chinese Academy of Sciences. He received the B.S. degree in applied mathematics and the M.S. degree in management science and engineering both from National University of Defense Technology, Changsha, China, the Ph.D. degree the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1998, 2003 and 2009, respectively. His research interests include handwriting recognition and ink document analysis.

Yu Shi is the director of Center on Research of Intelligent Security Technology, CIGIT, Chinese Academy of Sciences. He leads a team for core technology and industrial application research in Computer Vision and Pattern Recognition area. He has published more than 20 patents, and has obtained 4 patent licenses. He is the West Light A Class awarded by Chinese academy of sciences.

Steve Maybank received the BA degree in mathematics from King's College Cambridge in 1976 and the Ph.D. degree in computer science from Birkbeck College, University of London in 1988. He is currently a professor in the Department of Computer Science and Information Systems at Birkbeck College, University of London. His research interests include camera calibration, visual surveillance, etc. He is a Fellow of the IEEE and a Fellow of the Royal Statistical Society.