Efficient and Accurate Face Alignment by Global Regression and Cascaded Local Refinement

Jinzhan Su
Meitu, Inc
sjz@meitu.com

Zhe Wang
Meitu, Inc
wz@meitu.com

Chunyuan Liao
Hiscene Technology
liaocy@hiscene.com

Haibin Ling
Temple University
hbling@temple.edu

Abstract

Despite great advances witnessed on facial image alignment in recent years, high accuracy high speed face alignment algorithms still have rooms to improve especially for applications where computation resources are limited. Addressing this issue, we propose a new face landmark localization algorithm by combining global regression and local refinement. In particular, for a given image, our algorithm first estimates its global facial shape through a global regression network *(GRegNet) and then using cascaded* local refinement networks *(LRefNet) to sequentially improve the alignment result. Compared with previous face alignment algorithms, our key innovation is the sharing of low level features in GRegNet with LRefNet. Such feature sharing not only significantly improves the algorithm efficiency, but also allows full exploration of rich locality-sensitive details carried with shallow network layers and consequently boosts the localization accuracy. The advantages of our algorithm is clearly validated in our thorough experiments on four popular face alignment benchmarks, 300-W, AFLW, COFW and WFLW. On all datasets, our algorithm produces state-of-the-art alignment accuracy, while enjoys the smallest computational complexity.*

1. Introduction

Automatic face alignment from images, typically formulated as to face landmark localization, is critical in many computer vision and computer graphics applications, such as face recognition and verification [56, 38], face attribute retrieval [49], face animation [6] and face reenactment [39]. These tasks rely on locations of facial landmarks to spatially align the face, to predict the head pose or to fit 3D morphable models, and their performance directly depends on the accuracy of face alignment. Despite great advances witnessed in facial image alignment in recent years, however, high quality high speed face landmark localization algorithms still have large rooms to improve, especially for tasks where computation resource is limited, such as in mobile applications.

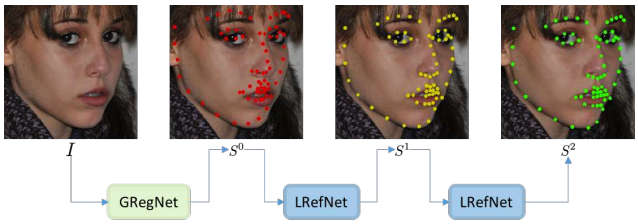


Figure 1. We propose a high accuracy low complexity image alignment algorithm. It first use a global regression (GRegNet) to initialize the landmark detection. Then, the result is sequentially improved by a cascaded set of local refinements (LRefNets). With efficient feature sharing and compact architecture design, our method compares favorably with recent state-of-the-arts in both model complexity and run time complexity, while generating best overall accuracy performance in our thorough experiments.

Among conventional (a.k.a. non-deep learning) face alignment methods, cascaded regression algorithms [7, 5, 22, 46] have achieved excellent performances when given a decent initialization. However, their performance often depends sensitively on initialization quality. Encouraged by recent popularity of deep learning models in computer vision, researchers have investigated using deep neural networks for initial shape estimator [50] or stage regressor [50, 40], leading to great improvement in alignment accuracy. Existing methods usually use separate models to estimate initial shape and regress shape details incrementally in consecutive stages. Such multi-stage strategy [14, 47, 37, 13, 29, 8] often boosts location accuracy of the initial result, while requests recomputing features at each stage. As a result, these solutions may have problems to achieve simultaneously high alignment accuracy and high run time efficiency.

Motivated by the above mentioned studies and meanwhile to address the issues, in this work we propose to combine the initial shape estimation and local refinement in a unified framework. Our key observation is that, a global shape regression network, in addition to provide ro-

bust global face localization, also carries rich localization information in its shallow network layers. In particular, intermediate feature maps from shallow layers preserve facial structure and have reasonable abstraction in the meantime (Figure 2). Following the observation, our solution is composed of two closely coupled components, namely a *global regression network* (GRegNet) and a few cascaded *local refinement networks* (LRefNets). Given an input facial image, GRegNet first holistically estimates all facial landmarks; then, starting from the output of GRegNet, LRefNets refines locally and incrementally each individual landmark. Different than previous solutions that request expensive feature extraction for similar refinement steps, our LRefNets reuse the low-level features extracted by GRegNet. More specifically, each LRefNet takes one shallow layer output of GRegNet, and several LRefNets are cascaded to explore coarse-to-fine layer outputs of GRegNet. The framework of the proposed algorithm is summarized in Figure 3.

In summary, our main contribution is a high quality high efficiency face landmark localization framework, with a novel feature sharing strategy between global shape regression and local landmark refinement. Such feature sharing not only significantly improves the algorithm efficiency, but also allows full exploration of rich localization-sensitive information carried with shallow network layers and consequently boosts the localization accuracy.

To empirically show the advantages of our algorithm, it is evaluated thoroughly on four popular face alignment benchmarks including the 300-W dataset [35], COFW-68 [16], AFLW [23] and WFLW [43]. On all datasets, our algorithm shows clear benefits over its baseline and produces state-of-the-art alignment accuracies. Moreover, our algorithm runs significantly faster than previously proposed algorithms, making it suitable for applications where computation resource is limited.

In the rest of the paper, we first summarize related work in Section 2. Then we introduce in details the proposed framework in Section 3, and present the experiment validation in Section 4. Finally, we draw the conclusion in Section 5.

2. Related work

Facial alignment, as a critical step for subsequent face analyses [56, 38, 49, 6, 39], has been intensively researched for many decades and impressive progress has been achieved. This is partially due to increasing data availability and variability [4, 27, 55, 34, 35, 36, 43], and due to advanced learning techniques that are tailored for face alignment and benefit from the data. Classical methods, such as ASMs [10, 31], AAMs [9, 41, 18, 1, 42] and CLMs [2, 3, 26] can hardly generalize in the wild, and the iteratively fitting is considerably expensive. Recently, cascaded regression methods and deep multi-stage methods are developed to achieve the state-of-the-art performance.

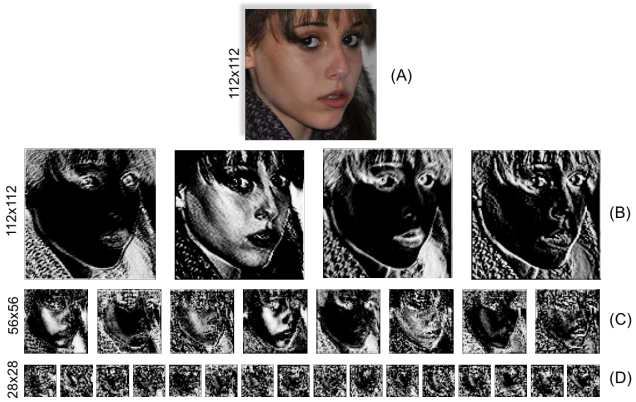


Figure 2. Visualization of shallow feature maps of GRegNet. Feature map normalization and histogram equalization are conducted for better visualization. (A) is input image of size 112×112; (B), (C) and (D) are feature maps generated from the first three convolutional or residual blocks respectively. Early feature maps (B, C) preserve facial structures and have proper abstractions of faces. As the layer goes deeper(D), abstraction get higher, and local detail information get lost.

2.1. Cascaded regression methods

Regression from image features to face shape in one step is extremely challenging. Cascaded regression methods divide the regression process into stages and learn shape increment at each stage. Conventional cascaded regression methods differ in the form of stage regressors, such as random ferns [7, 5], ensemble regression trees [22], Gaussian process regression tree (GPRT) [28] or linear regressor [46], and in different feature mapping functions, such as pixel different feature [7], hand-crafted SIFT [46], local binary feature [33]. Their performance strongly depends on the quality of shape initialization, the capability of stage regressor and the capacity of feature representation. Encouraged by deep learning methods, deep neural networks were used as initial shape estimator [50] or stage regressors [50, 40], leading to great improvement in alignment accuracy. Current methods usually use separate models to estimate initial shape and regress shape increment, and each stage regressor refines shape incrementally from image patches and hence performs feature extraction at each stage. Cascaded regression is also used in [21] for 3D-2D projection estimation to assist face landmark localization.

2.2. Multi-stage methods

Besides cascaded regression, multi-stage strategy is a common strategy to improve the accuracy of the initial result. The transformation stage [47, 8] or coarse shape prediction stage [14, 24, 29] are typically first used to warp the input image to a canonical pose. Then, the following stages

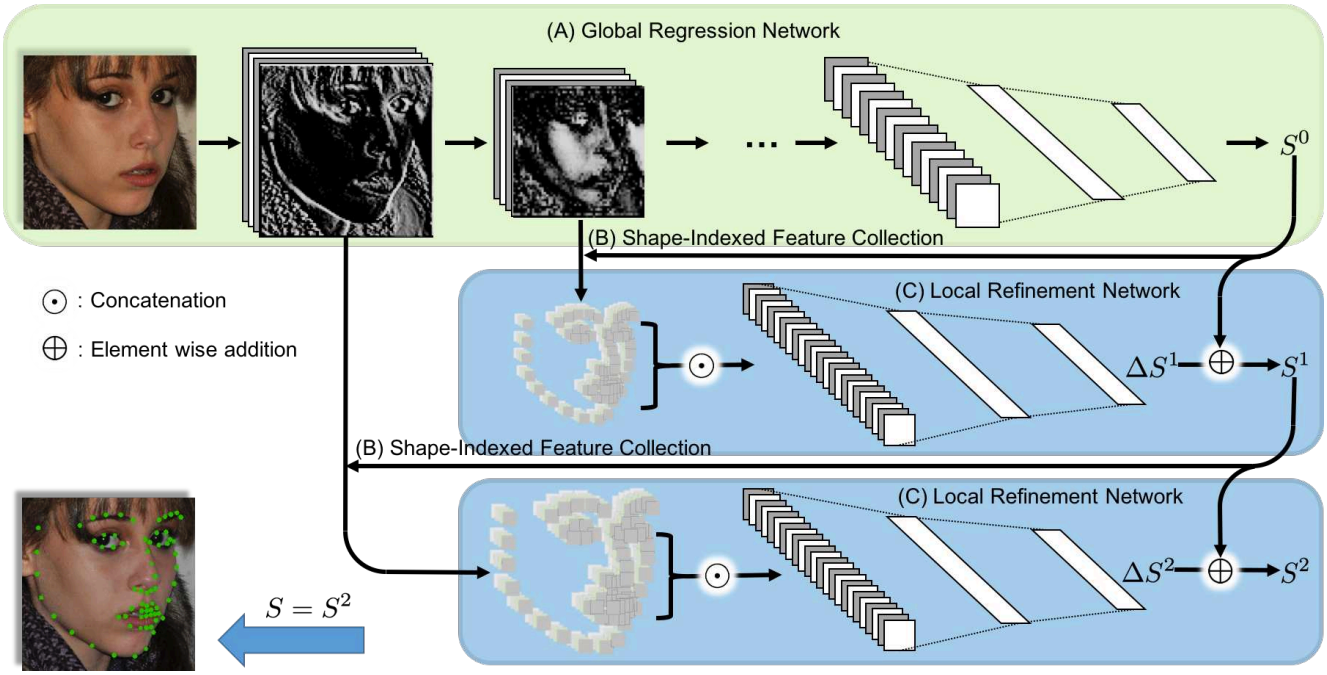


Figure 3. An overview of the proposed method with two LRefNets. (A) The input image I is first fed into the *Global Regression Network* (GRegNet) to estimate holistically an initial shape S^0. (B) Proposed *Shape-indexed feature collection* (SIFC) process is applied to crop features from feature maps of various scales preserved in GRegNet, based on the landmark locations provided by the shape out from previous stage (*i.e.*, S^0, S^1). (C) On the collected features, the *Local Refinement Networks* (LRefNets) sequentially improve the results by predicting shape increments (ΔS^1, ΔS^2) in the coarse-to-fine order. The result of the finest LRegNet is treated as the system output (*i.e.*, $S = S^2$).

perform fine-grained landmark localisation. To further improve accuracy, component-wise [8, 29] or point-wise [13] detection may also be conducted. Current multi-stage methods perform alignment from images and hence request recomputing features at each stage.

In this work, we proposed a unified face alignment framework that closely share features between global regression network (GRegNet) and cascaded local refinement networks (LRefNet). LRefNets reuse early intermediate features of GRegNet to refine the global estimation. Empirical experiments indicate that our method is efficient and accurate.

3. Proposed Method

3.1. Problem formulation and system overview

Given an input facial image I, our task is to localize a set of n predefined facial landmarks, denoted by $S \in \mathbb{R}^{2n}$. The proposed method addresses that by two main steps, namely *global regression network* (GRegNet), denoted by $\mathcal{G}$, and T cascaded *local refinement networks* (LRefNet), denoted by $\mathcal{R}^t, t = 1, \ldots, T$.

Specifically, the input image I is first fed into GRegNet to estimate, holistically, an initial shape (*i.e.*, set of landmarks) $S^0 \in \mathbb{R}^{2n}$. Meanwhile, T shallow layer feature maps of GRegNet, namely F^t, $t = 1, \ldots, T$ are preserved such that the T-th feature map corresponds to the shallowest layer (*i.e.*, with most details). Then, cascaded from coarse ($t = 1$) to fine ($t = T$), localized features for each landmark are extracted from F^t guided by the shape S^{t-1} from previous stage. Such feature reusing process, named *shape-indexed feature collection* (SIFC) and denoted by $\Phi(F^t)$, provides input for LRefNet $\mathcal{R}^t$. Then, $\mathcal{R}^t$ produces shape refinement vector ΔS^t, which is then combined with S^{t-1} to get an improved shape estimation S^t. Finally, the result of the finest LRegNet is treated as the system output, *i.e.* $S = S^T$.

The pipeline of our method is summarized in Figure 3 for $T = 2$. The pipeline can also be summarized by the following equations sequentially:

$$\{S^0, \{F^t\}_{t=1}^T\} = \mathcal{G}(I), \quad (1)$$

$$\Delta S^t = \mathcal{R}^t\big(\Phi(F^t, S^{t-1})\big), \quad (2)$$

$$S^t = S^{t-1} + \Delta S^t, \quad t = 1, \ldots, T \quad (3)$$

$$S = S^T. \quad (4)$$

The rest of this section describes the details of all the components in the pipeline as well as the training of the models.

Table 1. Architecture of GRegNet, which is based on *ResNet-18* [20], with following differences: conv1 has a kernel of 3×3 and stride 1; max pooling is removed; global average pooling is replaced by global depthwise convolution layer, namely gdconv in the table.

Name	Layer	Shape in
conv1	3×3, 64,S1,P1	$3 \times 112 \times 112$
res1	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$64 \times 112 \times 112$
res2	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$64 \times 56 \times 56$
res3	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 268 \end{bmatrix} \times 2$	$128 \times 28 \times 28$
res4	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$256 \times 14 \times 14$
gdconv	7×7	$512 \times 7 \times 7$
fc	-	1×512

* S: Stride, P: Padding, G: Group (same for Table 2)

3.2. Global regression network (GRegNet)

A high quality initial estimation of face shape is critical for the performance of a face alignment system. State-of-the-art face alignment algorithms mostly base themselves on deep neural networks with different variations for high localization precision. While it is tempting to borrow these algorithms directly for our GRegNet, the high computational cost force us to find cheaper solutions. Fortunately, as demonstrated in our thorough evaluation, by reusing intermediate features in GRegNet, our local refinement steps (LRefNets) successfully boost the final performance to state-of-the-arts, while being computationally very efficient.

In particular, our GRegNet $\mathcal{G}$ is modified from *ResNet-18* [20] followed by fully connected layers, which are adapted for the face alignment task. Details of the GRegNet architecture is given in Table 1.

Aside from the initial shape estimation S^0, $\mathcal{G}$ provides rich locality sensitive information to the following refinement by sharing with LRefNets shallow layer features $\{F^t\}_{t=1}^T$. In this work, to balance the run time efficiency and alignment accuracy, we choose two such layers, *i.e.*, $T = 2$. In particular, F^1 and F^2 are taken from the outputs of the *res1 layer* and the *conv1 layer* of GRegNet, respectively. These features, as shown in Figure 2, capture details of facial characteristics that can greatly benefit local shape refinement.

3.3. Shape-indexed feature collection (SIFC)

Intermediate features have been used popularly in vision tasks, and various methods were proposed to extract ROI features from feature maps such as in [17, 19]. The motivation for us to design shape-indexed feature collection for LRefNets is multi-fold: 1) for high localization quality, local features need to be extracted at subpixel precision; 2) the extraction process should be differentiable with respect to input features, so as to enable the loss of LRefNets to be back propagated to input layers, which is essential for the end-to-end training of both local refinement and global regression; and 3) the extraction should be efficient.

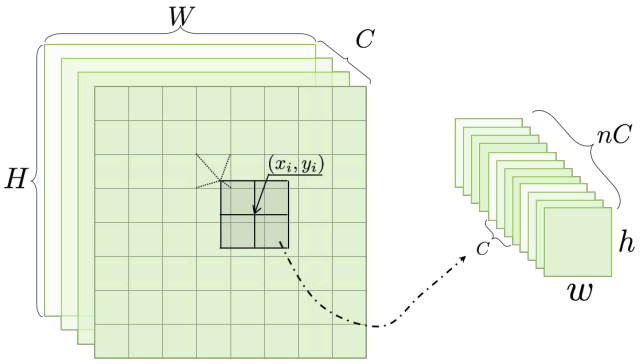


Figure 4. *Shape-indexed feature collection* (SIFC). At each rescaled landmark (x_i, y_i), a $w \times h$ patch centered at (x_i, y_i) are cropped from input feature maps with bilinear interpolation for achieving subpixel precision. Then n volume feature collected from n landmarks are concatenated to form the input of size $h \times w \times nC$ for LRefNet.

Table 2. Architecture details of LRefNet. C represents channel number of input feature maps, n is the number of landmarks.

Name	Layer	Shape in
Conv	5×5, $4n$, S1, P0, Gn	$nC \times 5 \times 5$
ReLU	-	$4n \times 1 \times 1$
FC	-	$1 \times 4n$

Thus motivated, we carefully design the *shape-indexed feature collection* (SIFC) process, denoted by $\Phi(F^t, S^{t-1})$, as illustrated in Figure 4. Specifically, let $F^t \in \mathbb{R}^{H \times W \times C}$ contains C channels of spatial resolution (W, H), and $S^{t-1} \in \mathbb{R}^{2n}$ contains coordinates of n landmark points at scale $t-1$. For each landmark point $\mathbf{p}'_i = (x', y')$, such that $x'_i = S^{t-1}(2i-1)$ and $y'_i = S^{t-1}(2i)$, $i = 1, 2, \ldots, n$, the goal is to collect features of C channels with spatial resolution (w, h), by cropping from corresponding location in F^t. For this purpose, we first rescale $\mathbf{p}'_i$ to $\mathbf{p}_i = (x_i, y_i)$ to match the spatial dimension of F^t. Then, for each channel $c \in \{1, \ldots, C\}$, we crop from the c-th channel of F^t a $w \times h$ patch centered at $\mathbf{p}_i$ by linear interpolation for achieving subpixel precision. Finally, all local features for each landmark form the output feature volume of size $h \times w \times nC$, which serves as the input for the local refinement module $\mathcal{R}^t$.

Aside from providing efficient local features, SIFC is naturally differentiable with respect to the input feature. This makes the whole system end-to-end trainable as described in Section 3.5. In our implementation we set $w = 5$ and $h = 5$ throughout all experiments.

3.4. Local refinement network (LRefNet)

Starting from the initial shape S^0 estimated by GRegNet, a sequence of T LRefNets refine the result in a cascaded way. To take full benefit of reusing shallow features shared by GRegNet through SIFC, LRefNets are designed to be very efficient, both in speed and in the number of parameters.

In particular, each LRefNet contains only one convolutional layer (including ReLU), followed immediately by a linear output layer. Furthermore, the convolution operations are grouped to run on each landmark independently, further reducing the computational complexity, and allowing convolutions focus on shape-indexed local features. Details of the LRegNet architecture is given in Table 2.

3.5. Optimization and training

Let $\theta = \theta_{\mathrm{G}} \bigcup \theta_{\mathrm{R}}$ be the set of all parameters in our model, where θ_{G} and θ_{R} denote sets of parameters for GRegNet and LRefNets, respectively. For an input image I with ground truth face shape S^* and estimated shape S, θ is typically learned through minimizing the loss function as for conventional face alignment networks:

$$\theta = \operatorname*{argmin}_{\theta} \mathcal{L}(S, S^*) \tag{5}$$

Note that in practice the loss is defined over the set of training samples, but in this section we focus on just one sample for notation conciseness.

In the ideal case, one can train the whole network by the above optimization. In practice, however, it is hard to train both θ_{G} and θ_{R} in one shot, especially considering that θ_{R} contains parameters for cascaded stages. Considering the fact that LRefNets are used to improve the result from GRegNet, it is natural to design a two-phase training process, such that the first one learns all parameters, while the second one fine tunes those in LRefNets.

Specifically, in phase one, we train θ by minimizing the following cost

$$\theta = \operatorname*{argmin}_{\theta} \left[\mathcal{L}_{\mathrm{G}}(S^0, S^*) + \sum_{t=1}^{T} \mathcal{L}_{\mathrm{R}}(\Delta S^t, \Delta S^{t^*})\right] \tag{6}$$

where ΔS^t is output of the t-th LRefNet, and ΔS^{t^*} is the target of ΔS^t which is calculated online during training

$$\Delta S^{t^*} = S^* - S^{t-1} \tag{7}$$

We use ℓ^2 loss for both cost functions $\mathcal{L}_{\mathrm{G}}$ and $\mathcal{L}_{\mathrm{R}}$.

Then, in the second phase, we fine tune LRefNets by the following optimization

$$\theta_{\mathrm{R}} = \operatorname*{argmin}_{\theta_{\mathrm{R}}} \sum_{t=1}^{T} \mathcal{L}_{\mathrm{R}}(\Delta S^t, \Delta S^{t^*}) \tag{8}$$

Details of training settings are discussed in Section 4. The training pipeline is summarized in Algorithm 1.

Algorithm 1 The training pipeline of proposed method

Input: Training data and network hyperparameters
Output: Trained network $\mathcal{G}$ and $\mathcal{R}^t, t = 1, \ldots, T$

// Phase 1
while training jointly **do**
  Forward as Equations 1–3
  $\Delta S^{t^*} \leftarrow S^* - \Delta S^{t-1}$ as Eq. 7
  $\ell \leftarrow \mathcal{L}(S^0, S^*) + \sum_{t=1}^{T} \mathcal{L}(\Delta S^t, \Delta S^{t^*})$
  Optimize $\mathcal{G}$ and $\{\mathcal{R}^t\}_{t=1}^{T}$ by minimize ℓ
end while

// Phase 2
while fine tune LRefNets **do**
  Forward as Equations 1–3
  $\Delta S^{t^*} \leftarrow S^* - \Delta S^{t-1}$ as Eq. 7
  $\ell \leftarrow \sum_{t=1}^{T} \mathcal{L}(\Delta S^t, \Delta S^{t^*})$
  Fine tune $\{\mathcal{R}^t\}_{t=1}^{T}$ by minimize ℓ
end while

4. Experiments

4.1. Datasets

300-W. The 300-W dataset is a combination of five face datasets including LFPW [4], AFW [55], HELEN [27], XM2VTS [30] and IBUG [36]. Images in the dataset have been semi-automatically annotated with 68 facial landmarks. Following the protocol in previous work [52], all the training samples from LFPW, HELEN and the full set of AFW are used as the training set, which contains 3,148 training samples. The common test set is formed by testing images from LFPW and HELEN, which contains 554 images. In addition, 135 samples in IBUG are regarded as the challenging testing subset. All testing samples form the full testing set with 689 samples.

COFW. For the COFW [5] dataset, we use the re-annotated COFW test set introduced by [16] to evaluate our algorithm.

AFLW. The AFLW dataset was first introduced in [23]. We evaluate the proposed method using the AFLW-Full protocol [53]. The original dataset provides up to 21 landmarks coordinates for each face but excluding invisible ones. The AFLW-Full dataset excludes ear landmarks and keeps other 19 landmarks, and invisible landmarks have been added manually. We use the data split introduced in [53]. Overall, 20,000 samples are used as the training set, and 4,386 samples as the test set.

WFLW. The WFLW dataset was introduced in [43]. It contains 10,000 faces with 98 manually annotated landmarks. We follow the protocol in [43] by using 7,500 samples of the dataset as the training set and the rest 2,500 samples as the test set. The test set is divided into 6 subsets by attributes, i.e, occlusion, pose, make-up, illumination, blur

and expression. The dataset introduces large variations in expression, pose and occlusion.

4.2. Evaluation metric

Normalized Mean Error (NME) and *Cumulative Errors Distribution* (CED) are used to evaluate the performance of proposed method. For the WFLW dataset, we follow [23] and use face size as the normalizing factor. For other datasets by default, errors are normalized by "inter-ocular" distance [35]. To better compare with previous results that are normalized by "inter-pupil" distance on 300-W dataset, we also report our results with "inter-pupil" normalization as shown in Table 3. In particular, we use the location averaged over all six points around eyes to get the position of pupils. In addition to NME and CED, the *Area Under the Curve*(AUC) and the failure rate at maximum error of 0.1. are also reported.

4.3. Implementation details

Before cropping, we pad the provided bounding box on 300-W, WFLW, and COFW. On the 300-W and COFW datasets, bounding boxes are padded by 10% isotropically. On WFLW, bounding boxes are padded with left 10%, top -12.3%, right 10% and bottom 7.7%, which is the average displacement between detected and ground truth bounding boxes on the WFLW training set. On the ALFW dataset, original bounding boxes are used. Then, all images are cropped and resized to 112×112 according padded bounding boxes.

Data Augmentation. Various augmentation techniques are used in this work. We randomly translate each training sample by 10% of the size of its bounding box. Samples are randomly rotated between $[-30, 30]$ degrees, and bounding boxes are randomly scaled in range $[0.75, 1.2]$. In addition, samples are randomly flipped horizontally with probability of 50%. We use *imgaug* package[1] to perform pixel oriented augmentation. *Gaussian blur, average blur, median blur, sharpen, emboss, additive Gaussian noise, add, add to hue and saturation, contrast normalization*, and *gray scale* are applied. To balance the pose distribution, similar techniques to PDB [14] are applied to the 300-W training set. Except for pose balancing, we perform augmentation online during training and generate random samples in every epoch.

Training We use *PyTorch* [32] for all experiments. SIFC is implemented as a common module on the *PyTorch* framework. We use the SGD optimizer with base learning rate 0.01, momentum 0.9, weight decay 5e-4. *ReduceLROnPlateau* scheduler is used to adjust the learning rate configured with patience 200 epochs, learning rate decay factor is set to 0.75. As described in Algorithm 1, we train the networks in two phases. In phase one, GRegNet and LRefNets are trained jointly up to 3,000 epochs. In phase two, we reduce base learning rate to 0.001 and fine tune LRegNets for 1,500 epochs. In our experiments, we find that GRegNet fit 300-W training set so well that makes ΔS^{t^*} (Eq. 7) insignificant for following LRefNets to learn the incremental refinement. To address this problem, dropout is added to GRegNet in phase two when generating initial shape estimation. For fair evaluation, no extra data is used in all experiments, and all models are trained from scratch.

[1] https://github.com/aleju/imgaug

Table 3. A Comparison on 300-W. The three best scores are indicated in **red**, **green** and **blue**, respectively. Best viewed in color.

	Method	300-W		
		Common	Challenging	Fullset
Inter-pupil normalization	RCPR [5]	6.18	17.26	8.35
	CFAN [50]	5.5	16.78	7.69
	ESR [7]	5.28	17	7.58
	SDM [46]	5.57	15.4	7.5
	LBF [33]	4.95	11.98	6.32
	CFSS [52]	4.73	9.98	5.76
	3DDFA [54]	6.15	10.59	7.01
	MDM [40]	4.83	10.14	5.88
	DVLN [44]	3.94	7.62	4.66
	LAB [43]	3.42	6.98	4.12
	Wing(CNN6/7) [14]	**3.27**	7.18	**4.04**
	baseline(GRegNet)	4.11	7.32	4.74
	Ours	3.76	**6.89**	4.37
Inter-ocular normalization	TCDCN [51]	4.8	8.6	5.54
	Two-Stage [29]	4.36	7.56	4.99
	RAR [45]	4.12	8.35	4.94
	PCD-CNN [25]	3.67	7.62	4.44
	SAN [11]	3.34	6.6	3.98
	SBR [12]	3.28	7.58	4.1
	LAB [43]	2.98	5.19	3.49
	baseline(GRegNet)	2.97	5.07	3.38
	Ours	**2.71**	**4.78**	**3.12**

4.4. Results and discussion

Comparison with state-of-the-arts. On 300-W, as reported in Table 3, our method achieves excellent results that are better than or similar to state-of-the-arts. Our method generates the best scores in Inter-ocular NME on all three subsets. For Inter-pupil NME, our method also performs among the bests, especially on the challenging subset. On the AFLW dataset, as shown in Table 5, our method reaches the state-of-the art performance and ranks the second in both full and frontal settings, while runs much more efficient than the champion. These results show clearly the benefit brought by our concise and efficient combination of global regression and local refinement, as well as the feature sharing between them.

Our method continues generating excellent results on WFLW and COFW-68 by generating new state-of-the-art results. On WFLW, following the work in [43], we report mean error, failure rate and AUC on the testset and six sub-

Table 4. Evaluation on WFLW in terms of the NME. The baseline results are estimated by GRegNet. The three best scores are indicated in **red**, **green** and **blue**, respectively. Best viewed in color.

Metric	Method	FullSet	Pose	Expression	Illumination	Makeup	Occlusion	Blur
NME(%)	ESR [7]	11.13	25.88	11.47	10.49	11.05	13.75	12.20
	SDM [46]	10.29	24.10	11.45	9.32	9.38	13.03	11.28
	CFSS [52]	9.07	21.36	10.09	8.30	8.74	11.76	9.96
	DVLN [44]	6.08	11.54	6.78	5.73	5.98	7.33	6.88
	LAB [43]	5.27	10.24	5.51	5.23	5.15	6.79	6.32
	Wing(ResNet-50) [14]	5.11	8.75	5.36	4.93	5.41	6.37	5.81
	baseline(GRegNet)	5.09	8.71	5.62	4.91	5.22	6.09	5.63
	Ours	**4.65**	**7.99**	**5.13**	**4.49**	**4.74**	**5.67**	**5.24**
Failure Rate (%)	ESR [7]	35.24	90.18	42.04	30.80	38.84	47.28	41.40
	SDM [46]	29.40	84.36	33.44	26.22	27.67	41.85	35.32
	CFSS [52]	20.56	66.26	23.25	17.34	21.84	32.88	23.67
	DVLN [44]	10.84	46.93	11.15	7.31	11.65	16.30	13.71
	LAB [43]	7.56	28.83	6.37	6.73	7.77	13.72	10.74
	Wing(ResNet-50) [14]	6.00	22.70	**4.78**	**4.30**	7.77	12.50	7.76
	baseline(GRegNet)	6.44	25.77	6.37	5.73	5.83	11.96	8.02
	Ours	**4.88**	**18.40**	**4.78**	4.44	**5.34**	**9.65**	**6.99**
AUC@0.1	ESR [7]	0.2774	0.0177	0.1981	0.2953	0.2485	0.1946	0.2204
	SDM [46]	0.3002	0.0226	0.2293	0.3237	0.3125	0.2060	0.2398
	CFSS [52]	0.3659	0.0632	0.3157	0.3854	0.3691	0.2688	0.3037
	DVLN [44]	0.4551	0.1474	0.3889	0.4743	0.4494	0.3794	0.3973
	LAB [43]	0.5323	0.2345	0.4951	0.5433	0.5394	0.4490	0.4630
	Wing(ResNet-50) [14]	0.5504	0.3100	0.4959	0.5408	0.5582	0.4885	0.4918
	baseline(GRegNet)	0.5485	0.3172	0.4919	0.5539	0.5332	0.4958	0.5074
	Ours	**0.5824**	**0.3570**	**0.5211**	**0.5921**	**0.5667**	**0.5250**	**0.5399**

Table 5. A comparison on AFLW dataset in terms of NME. The three best scores are indicated in **red**, **green** and **blue**, respectively. Best viewed in color.

Method	AFLW-Full(%)	AFLW-Frontal(%)
CDM [48]	5.43	3.77
RCPR [5]	3.73	2.87
ERT [7]	4.35	4.35
LBF [33]	4.25	2.74
CFSS [52]	3.92	2.68
CCL [53]	2.72	2.17
TSR [29]	2.17	-
DAC-OSR [15]	2.27	1.81
Wing(CNN6/7) [14]	1.65	-
LAB [43]	**1.25**	**1.14**
baseline(GRegNet)	1.89	1.78
Ours	**1.63**	**1.46**

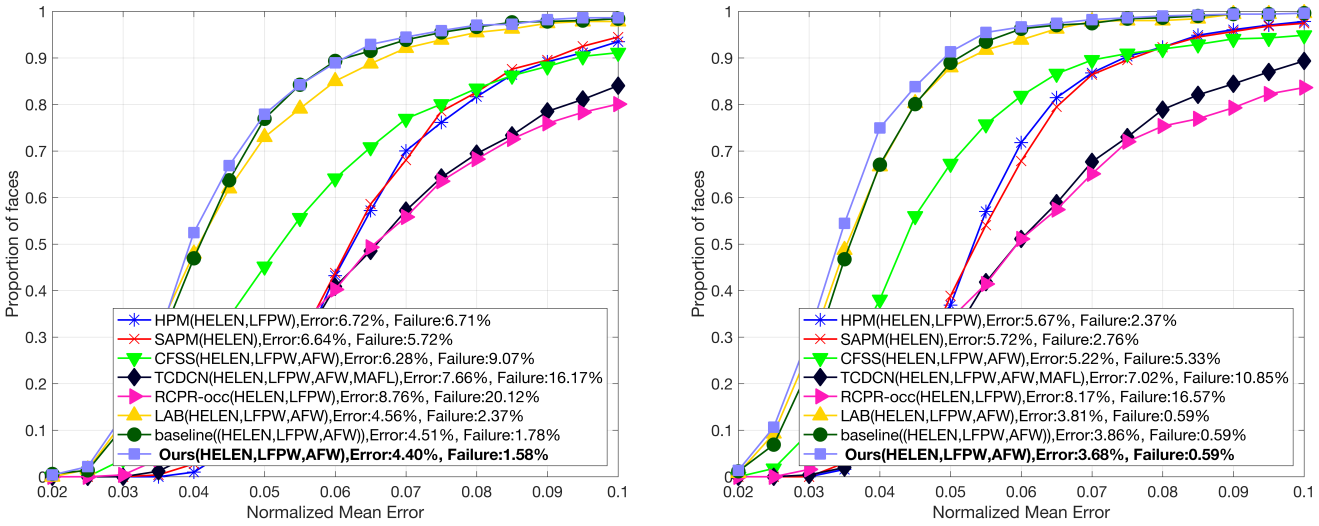
sets, as summarized in Table 4. Our method outperforms all previous state-of-the-art methods (as reported on github[2] by the author of [14]) by a large margin. We further reduce mean error on the full set from 5.11% to 4.65%, *i.e.*, 9.0% relative performance improvement. The proposed method achieves best performances in all criteria on all subsets, except for a second place in the failure rate on the illumination one.

COFW-68 is a re-annotated version of the COFW test following the same annotation protocol as that for 300-W. We conduct evaluation on COFW-68 using the same model trained on 300-W dataset. Two aspects of the accuracy is evaluated based on the manually annotated visibility attributes of landmarks. The comparisons are measured by the CED curves, as plotted in Figure 5a and 5b. Our method achieves the best accuracy among the state-of-the-art methods with mean error 4.40% for all landmarks and 3.68% for visible landmarks, compared to previous best [43] with 4.58% and 3.81% mean error.

Aside from achieving the above highly accurate results, it is worth noting that, our method does not sacrifices in the run time efficiency, nor in model complexity. This will be quantitatively analyzed later in this section.

Comparison with baseline. To help understand the benefit of the proposed two stage localization algorithm, we treat the GRegNet as a baseline algorithm, *i.e.*, no local refinement. Its performance on 300-W is shown in Table 3 along with results from other methods. From the table we see that, on the Fullset, GRegNet achieves accuracies of 4.74 (inter-pupil NME) and 3.38% (inter-ocular NME), both of which are clearly lower than the proposed final solution (4.37% and 3.12% respectively). Similar observations can be found on other settings as well as other datasets as well, *i.e.*, see Table 4 for WFLW, Table 5 for AFLW, and Figures 5a and 5b) for COFW. These results convincingly show the clear

[2] https://github.com/FengZhenhua/Wing-Loss



(a) Evaluation on all landmarks

(b) Evaluation on visible landmarks

Figure 5. CED for the COFW-68 testset. For each method, the training set (in parentheses), mean error and failure rate are summarized in the legend.

Table 6. A comparison among face alignment methods on WFLW dataset in terms of network complexity.

Method	# Parameter(M)	# MACC(G)
DVLN [44]	269.65	46.40
Wing(ResNet-50) [14]	23.79	4.09
LAB [43]	12.22	18.70
baseline(GRegNet)	11.56	1.7430
Ours	12.97	1.7444

benefits of using LRefNets to refine the initial result generated by GRegNet.

Network complexity. As discussed in previous sections, the main motivation for our method is to design a landmark localization algorithm that not only generates high precision results but also with low complexity, so as to facilitate the deployment to resource limited scenarios such as mobile applications. Both GRegNet and LRefNets are designed following this motivation, and uses concise structures that powerfully and effciently collaborate with each other.

In practice, algorithm efficiency depends on various factors. For objective evaluating the algorithm, we use the estimated number of parameters of network and number of Multiplication Accumulation operations (MACC) for measuring respectively the model complexity and the run time complexity.

The statistics of different algorithms compared on the WFLW dataset are summarized in Table 6. It clearly show the efficiency of our algorithm, especially the running time complexity. The table also shows that only 0.9‰ of MACCs of our model is due LRefNets, which is barely noticeable in practice, especially considering the significant accuracy improvement. In practice, on a PC (i7-4770 at 2.20GHz), our unoptimized implementation runs about 150ms per image.

5. Conclusion

In this work, we proposed a high accuracy high speed face alignment algorithm, which effectively combines global shape regression (*i.e*. GRegNet) and a cascaded set of local shape refinements (*i.e*. LRefNets). Through a carefully designed feature sharing strategy, the locality-sensitive shallow features generated by the GRegNet are efficiently shared with LRefNets, which then refines the initial output from GRegNet sequentially to high quality output. Aside from the feature sharing strategy, the carefully designed compact network architecture helps further boost the computational efficiency. In our thorough experiments on four popular face alignment benchmarks, our method produces state-of-the-arts accuracies, while enjoying low complexity in both computation and model parameters.

References

[1] E. Antonakos, J. Alabort-i Medina, G. Tzimiropoulos, and S. Zafeiriou. Hog active appearance models. In *ICIP*, 2014. 2

[2] T. Baltrušaitis, P. Robinson, and L.-P. Morency. 3d constrained local model for rigid and non-rigid facial tracking. In *CVPR*, 2012. 2

[3] T. Baltrusaitis, P. Robinson, and L.-P. Morency. Constrained local neural fields for robust facial landmark detection in the wild. In *ICCV Workshops*, 2013. 2

[4] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. *TPAMI*, 35(12), 2013. 2, 5

[5] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *ICCV*, 2013. 1, 2, 5, 6, 7

[6] C. Cao, Y. Weng, S. Lin, and K. Zhou. 3d shape regression for real-time facial animation. *ACM TOG*, 32(4), 2013. 1, 2

[7] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *CVPR*, 2012. 1, 2, 6, 7

[8] X. Chen, E. Zhou, Y. Mo, J. Liu, and Z. Cao. Delving deep into coarse-to-fine framework for facial landmark localization. In *CVPR Workshops*, 2017. 1, 2, 3

[9] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *TPAMI*, (6), 2001. 2

[10] T. F. Cootes and C. J. Taylor. Active shape modelssmart snakes. In *BMVC*. 1992. 2

[11] X. Dong, Y. Yan, W. Ouyang, and Y. Yang. Style aggregated network for facial landmark detection. In *CVPR*, 2018. 6

[12] 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. 6

[13] Y. Dong and Y. Wu. Adaptive cascade deep convolutional neural networks for face alignment. *Computer standards & interfaces*, 42, 2015. 1, 3

[14] 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. 1, 2, 6, 7, 8

[15] 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. 7

[16] G. Ghiasi and C. C. Fowlkes. Occlusion coherence: Detecting and localizing occluded faces. *arXiv preprint arXiv:1506.08347*, 2015. 2, 5

[17] R. Girshick. Fast r-cnn. In *ICCV*, 2015. 4

[18] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(12), 2005. 2

[19] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017. 4

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[21] A. Jourabloo, X. Liu, M. Ye, and L. Ren. Pose-invariant face alignment with a single cnn. In *ICCV*, 2017. 2

[22] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *CVPR*, 2014. 1, 2

[23] 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 *ICCV Workshops*, 2011. 2, 5, 6

[24] M. Kowalski, J. Naruniec, and T. Trzcinski. Deep alignment network: A convolutional neural network for robust face alignment. In *CVPR Workshops, Faces-in-the-wild Workshop/Challenge*, volume 3, 2017. 2

[25] A. Kumar and R. Chellappa. Disentangling 3d pose in a dendritic cnn for unconstrained 2d face alignment. In *CVPR*, 2018. 6

[26] N. Kumar, P. Belhumeur, and S. Nayar. Facetracer: A search engine for large collections of images with faces. In *ECCV*, 2008. 2

[27] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *ECCV*, 2012. 2, 5

[28] D. Lee, H. Park, and C. D. Yoo. Face alignment using cascade gaussian process regression trees. In *CVPR*, 2015. 2

[29] J.-J. Lv, X. Shao, J. Xing, C. Cheng, X. Zhou, et al. A deep regression architecture with two-stage re-initialization for high performance facial landmark detection. In *CVPR*, 2017. 1, 2, 3, 6, 7

[30] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. Xm2vtsdb: The extended m2vts database. In *AVBPA*, 1999. 5

[31] S. Milborrow and F. Nicolls. Locating facial features with an extended active shape model. In *ECCV*, 2008. 2

[32] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. *NIPS Workshop*, 2017. 6

[33] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *CVPR*, 2014. 2, 6, 7

[34] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: Database and results. *Image and vision computing*, 47, 2016. 2

[35] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *ICCV Workshops*, 2013. 2, 6

[36] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *CVPR Workshops*, 2013. 2, 5

[37] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *CVPR*, 2013. 1

[38] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR*, 2014. 1, 2

[39] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, 2016. 1, 2

[40] 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. 1, 2, 6

[41] G. Tzimiropoulos, J. Alabort-i Medina, S. Zafeiriou, and M. Pantic. Generic active appearance models revisited. In *ACCV*, 2012. 2

[42] G. Tzimiropoulos and M. Pantic. Optimization problems for fast aam fitting in-the-wild. In *ICCV*, 2013. 2

[43] 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. 2, 5, 6, 7, 8

[44] W. Wu and S. Yang. Leveraging intra and inter-dataset variations for robust face alignment. In *CVPR Workshops*, 2017. 6, 7, 8

[45] 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. 6

[46] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, 2013. 1, 2, 6, 7

[47] J. Yang, Q. Liu, and K. Zhang. Stacked hourglass network for robust facial landmark localisation. In *CVPR Workshops*, 2017. 1, 2

[48] X. Yu, J. Huang, S. Zhang, W. Yan, and D. N. Metaxas. Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model. In *ICCV*, 2013. 7

[49] J. Zeng, S. Shan, and X. Chen. Facial expression recognition with inconsistently annotated datasets. In *ECCV*, 2018. 1, 2

[50] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *ECCV*, 2014. 1, 2, 6

[51] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Learning deep representation for face alignment with auxiliary attributes. *TPAMI*, 38(5), 2016. 6

[52] S. Zhu, C. Li, C. Change Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *CVPR*, 2015. 5, 6, 7

[53] S. Zhu, C. Li, C.-C. Loy, and X. Tang. Unconstrained face alignment via cascaded compositional learning. In *CVPR*, 2016. 5, 7

[54] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3d solution. In *CVPR*, 2016. 6

[55] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012. 2, 5

[56] Z. Zhu, P. Luo, X. Wang, and X. Tang. Deep learning identity-preserving face space. In *ICCV*, 2013. 1, 2