The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Facial Landmarks Detection by Self-Iterative Regression Based Landmarks-Attention Network

Tao Hu,[1] Honggang Qi,[1] Jizheng Xu,[2] Qingming Huang[1]
[1] University of Chinese Academy of Sciences, Beijing, China
[2] Microsoft Research Asia, Beijing, China
hutao16@mails.ucas.ac.cn, hgqi@ucas.ac.cn

Abstract

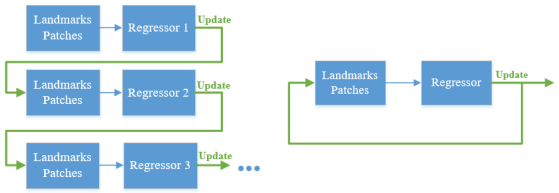
Cascaded Regression (CR) based methods have been proposed to solve facial landmarks detection problem, which learn a series of descent directions by multiple cascaded regressors separately trained in coarse and fine stages. They outperform the traditional gradient descent based methods in both accuracy and running speed. However, cascaded regression is not robust enough because each regressor's training data comes from the output of previous regressor. Moreover, training multiple regressors requires lots of computing resources, especially for deep learning based methods. In this paper, we develop a Self-Iterative Regression (SIR) framework to improve the model efficiency. Only one self-iterative regressor is trained to learn the descent directions for samples from coarse stages to fine stages, and parameters are iteratively updated by the same regressor. Specifically, we proposed Landmarks-Attention Network (LAN) as our regressor, which concurrently learns features around each landmark and obtains the holistic location increment. By doing so, not only the rest of regressors are removed to simplify the training process, but the number of model parameters is significantly decreased. The experiments demonstrate that with only 3.72M model parameters, our proposed method achieves the state-of-the-art performance.

Introduction

Facial landmarks detection is one of the most important techniques in face analysis, such as face recognition, facial animation and 3D face reconstruction. It aims to detect the facial landmarks such as eyes, nose and mouth, namely predicting the location parameters of landmarks.

Researchers usually regard this task as a typical non-linear least squares problem (Xiong and la Torre 2013). The Newton's method and its variants are the traditional gradient based solution, whose convergence rate is quadratic and is guaranteed to converge, provided that the initial estimate is sufficiently close to the minimum. However, when the objective function is not differentiable(*e.g.* SIFT(Lowe 2004)) or the Hessian matrix is not positive definite, the method won't works well (Xiong and la Torre 2013; 2015).

In recent years, cascaded regression based methods (Dollár, Welinder, and Perona 2010; Cao et al. 2014; Xiong and la Torre 2013; Ren et al. 2014; Zhu et al. 2016; Liu et al. 2016; Tzimiropoulos 2015; Tu 2008) have been proposed and applied to solve the non-linear least squares problem. They usually train multiple regressors to predict the parameters' increment sequentially, which outperform the traditional gradient descent based methods in both accuracy and running speed. Moreover, deep learning based cascaded regression methods (Sun, Wang, and Tang 2013; Zhang et al. 2014b; Trigeorgis et al. 2016; Xiao et al. 2016; Zhang et al. 2014a) are widely leveraged for this task because of the powerful ability to extract the discriminative feature. However, when applying cascaded regression system, three main problems arise: (1) Each regressor just works well in its local data space, when previous regressor predicts the false descent direction, the final results are very likely to drift away; (2) In general, higher accuracy can be obtained by adding more cascaded regressors, while it will increase model storage memory and computing resources; (3) Subsequent regressors usually cannot be activated for training until previous regressors finished their training process, which increases the system complexity.



(a) Cascaded Regression. (b) Self-Iterative Regression.

Figure 1: Facial landmarks detection process of Cascaded Regression(a) and Self-Iterative Regression(b). To predict the landmarks' location parameters, the CR based methods require multiple regressors, while SIR just need one regressor and updates parameters iteratively.

In this paper, we develop a Self-Iterative Regression (SIR) framework to solve the above issues. By means of the powerful representation of Convolutional Neural Network (CNN), we only train *one* regressor to learn the descent directions in coarse and fine stages together. The training data is obtained by random sampling in the parameter space, and in the testing process, parameters are updated iteratively by calling the

same regressor, which is dubbed Self-Iterative Regression. The testing process is illustrated in Figure 1(b). The experimental results show that for deep learning based method, one regressor achieves comparable performance to state-of-the-art multiple cascaded regressors and significantly reduce the training complexity. Moreover, to obtain discriminative landmarks features, we proposed a Landmarks-Attention Network (LAN), which focuses on the appearance around landmarks. It first concurrently extracts local landmarks' features and then obtains the holistic increment, which significantly reduces the dimension of the final feature layer and the number of model parameters. The contributions of this paper are summarized as follows:

1. We propose a novel regression framework called SIR to solve the non-linear least squares problem, which simplifies the cascaded regression framework and obtains state-of-the-art performance in facial landmarks detection task.
2. The Landmarks-Attention Network (LAN) is developed to independently learn discriminative features around each landmarks, which significantly reduces the dimension of feature layer and the number of model parameters.
3. Experimental results on several publicly available benchmarks demonstrate the effectiveness of the proposed method.

Related Work

In this section, we will review related works in solving non-linear least squares problems, especially facial landmarks detection problem.

Cascaded Regression based Methods. Cascaded regression was first introduced by Dollár *et al.* (Dollár, Welinder, and Perona 2010), which trains a fixed cascaded linear week regressors to predict the pose parameter of the object. Then, Xiong *et al.* describes the cascaded regression problem as a general learning framework called Supervised Descent Method (SDM) in (Xiong and la Torre 2013). It avoids computing the Jacobian and Hessian matrix by learning a sequence of local descent directions to minimize the non-linear least squares function. To accelerate the running speed of facial landmarks detection, LBF is developed in (Ren et al. 2014), which learns local binary feature with random forest(Breiman 2001) and obtains the final output by jointly learning the linear regression. To obtain a robust initialization, the CFSS (Zhu et al. 2015) first performs a coarse shape search over the shape spaces and then constrains the subsequent refinement regressors by the coarse shape.

Deep Learning based Methods. The CNN based methods can extract more discriminative features than above methods. Sun *et al.* (Sun, Wang, and Tang 2013) presents a deep cascaded regression based method by cascading three levels of CNNs and it regress the location of facial landmarks with the coarse-to-fine strategy. The disadvantage is obvious: too many CNNs (23 CNNs in their work) need to be trained, which requires too much computing resources.

Zhang *et al.* develops a Coarse-to-Fine Auto-encoder Network (CFAN) (Zhang et al. 2014a), which consists of multiple Stacked Auto-encoder Networks (SANs). The first SAN quickly predicts the preliminary location of landmarks by a low-resolution image, and the subsequent SANs then refine the location with higher and higher resolution.

Trigeorgis *et al.* proposed the Mnemonic Descent Method (MDM) (Trigeorgis et al. 2016), which regards the non-linear least squares optimization as a dynamic process. The Recurrent Neural Network (RNN) is introduced to maintain an internal memory unit that accumulates the history information so as to relate the cascaded refinement process.

João *et al.* proposed a iterative error feedback (Carreira et al. 2016) method to solve the human pose extimation problems. Same with MDM, their training data is generated by previous stages, while ours is obtained by random sampling in coarse stages and fine stages, which simplifies the training process.

Xiao *et al.* (Xiao et al. 2016) propose a Long Short Term Memory (LSTM) based recurrent attentive-refinement network, which also follows the pipeline of cascaded regressions. Instead of updating all landmarks location together, it first extracts reliable landmarks by a CNN and then infers locations of the rest noisy landmarks, resulting in improved accuracy. However, these deep cascaded regression methods usually require more computing resources and also suffer from the same drawbacks as discussed above.

Cascaded Regression

Before introducing our method, we begin with the cascaded regression framework in brief for better understanding. As illustrated in Figure 1(a), in the training process of cascaded regression, K regressors $(R_1, R_2, \cdots, R_K)$ are trained sequentially. Each regressor R_k is computed by minimizing the expected loss between the predicted and the optimal parameters's increment. It is formulated as

$$R_k = \underset{R_k}{\operatorname{argmin}} \sum_i \|\Delta\theta^*_{k,i} - R_k(x_{k,i})\|_2^2, k = 1, 2, \cdots, K, \tag{1}$$

where $x_{k,i}$ is i_{th} example in k_{th} regression process, $\Delta\theta^*_{k,i} = \theta^*_i - \theta_{k,i}$ is the corresponding target increment, *i.e.*, the difference between ground truth parameter θ^*_i and present parameter $\theta_{k,i}$. After obtaining R_k, the target parameter is updated by Equ. (2),

$$\theta_{k+1,i} = \theta_{k,i} + R_k(x_{k,i}). \tag{2}$$

Then, new training dataset will be generated according to the updated parameter for the next regression (Xiong and la Torre 2013).

In the testing process, parameter will be sequentially refined by these cascaded regressors in Equ 2.

Self-Iterative Regression

In this section, we will describe our facial landmarks detection method including the Gaussian random sampling and the Landmarks-Attention Network in detail. The overall procedure is presented in Figure 2.

Gaussian random Sampling

Generating training data is key important process in our method. Cascaded regression generates training data according to previous regressor, while our method obtains it

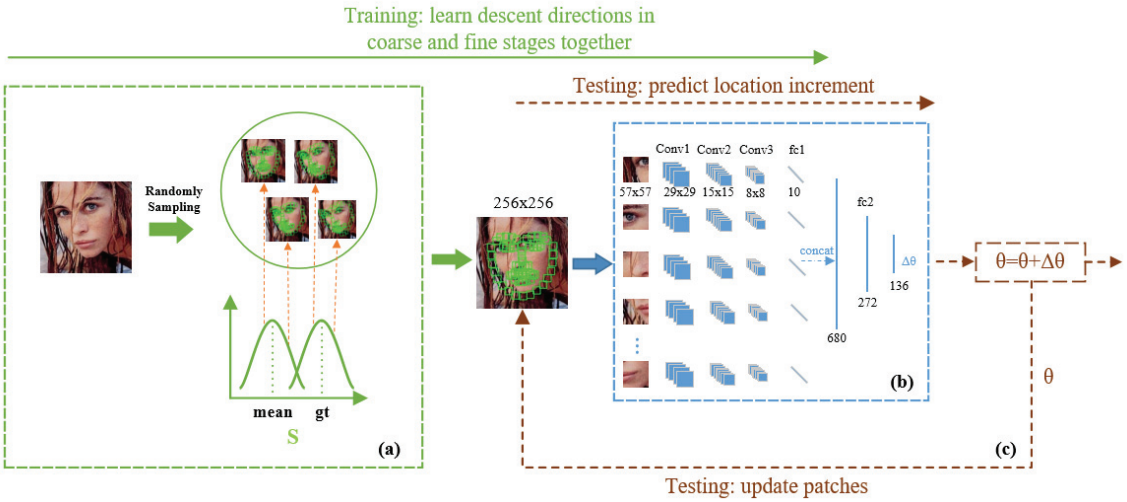


Figure 2: Training and testing process of the proposed SIR. (a) random sampling process. (b) Landmarks-Attention Network. (c) Iterative predicting and updating process. The training process consists of (a) and (b), while the testing process consists of (b) and (c). In the figure, one of the dimension of facial Landmarks Model parameter S is showed, and θ is landmarks' location parameter.

by random sampling, which includes most possible landmarks distribution from coarse stages to fine stages. Let (x_j, y_j) be the j_{th} landmark's position coordinates and $\theta = (x_1, y_1, x_2, y_2, ..., x_M, y_M)$ be all landmarks' location parameters, where M is the total number of facial landmarks. It is not a good choice to directly sample in location parameter θ since its dimension is so high that the training process will be hard to converge and is very likely to generate unreasonable face shape. To improve the effectiveness of sampling process, we indirectly obtain sampling location θ according to a new facial landmarks model that is similar to 3D Morphable Model (3DMM) (Blanz and Vetter 1999). Facial landmarks distribution will be represented by pose and shape parameter.

Facial Landmarks Model. We obtain intrinsic face shape parameter by Principal Component Analysis (PCA) and pose parameter(including 2D translation, in-plane rotation and scale) by geometry transformation. The shape, translation, rotation angle and scale coefficient are represented by α, t_{2d}, β, f respectively. Finally, facial landmarks model parameters can be represented by $S = [\alpha, t_{2d}, \beta, f]$. S and θ are two kinds of representation for facial landmarks. S can be converted to θ by

$$\theta(S) = f * R_\beta * (S_0 + A * \alpha) + t_{2d}, \tag{3}$$

where S_0 is the mean shape, A is the PCA shape matrix and $R_\beta = \begin{bmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{bmatrix}$ is the rotation matrix with angle β.

Random sampling in facial landmarks model S and then converting to location parameter θ makes the sampling process easier to control and generates more reasonable landmarks' distribution.

Sampling space. For each face I, let S_{gt} represent its ground truth facial landmarks model parameters. We random select values in each dimension of S obeying distribution D which is a union set of two Gaussian distribution. The sampling space of each face is represented by

$$D \sim \{N(S_0, \sigma) \cup N(S_{gt}, \sigma)\}, \tag{4}$$

where $N(\cdot, \cdot)$ represents Gaussian distribution, and σ is its standard deviation.

We adopt this sampling distribution because training regressor around mean location and ground truth location affects the performance in coarse and fine stages, respectively, and the final location error usually obeys Gaussian distribution. The value of standard deviation σ affects the final performance. System with larger σ will contain more training space which makes the system more robust, while the final accuracy may decrease because sampling probability around ground truth will decrease and vise versa. The effect of σ will be discussed in the *Experiments* section.

For i-th image in the t-th sampling period, sampling parameter $S_{t,i}$ is obtained by random selecting a value in Equ. (4). We then calculate location parameters $\theta_{t,i}$ by Equ (3) and extract patches $P_{t,i}$ in location $\theta_{t,i}$. Finally, we set $P_{t,i}$ as the training input data and set $\Delta\theta_{t,i} = \theta_i^* - \theta_{t,i}$ as regressor's corresponding target increment. The process is also summarized in Algorithm 1, and the training data is represented as

$$\bigcup_{t=1}^{T} \bigcup_{i=1}^{N} (P_{t,i}, \Delta\theta_{t,i}), \tag{5}$$

where T is the number of sampling period, N is the number of images in raw dataset.

The sampling process is illustrated in Figure 2(a). By the sampling process, we obtained nearly unlimited training data and the training space contains most possible landmarks' distribution from coarse stages to fine stages. The sampled training data is online generated to save the system memory.

Algorithm 1 Sampling process of SIR

Input: Raw face landmarks dataset: $\bigcup_{i=1}^{N}(I_i, (S_{gt})_i)$

Output: Training dataset: $\bigcup_{t=1}^{T}\bigcup_{i=1}^{N}(P_{t,i}, \Delta\theta_{t,i})$

for $t = 1$ to T **do**
  for $i = 1$ to N **do**
    random select value $S_{t,i}$ in Equ. (4);
    Calculate location $\theta_{t,i}$ by $S_{t,i}$ by Equ. (3);
    Extract patches $P_{t,i}$ for image I_i in location $\theta_{t,i}$;
    Set $P_{t,i}$ as the regressor's input data;
    Set $\Delta\theta_{t,i} = \theta_i^* - \theta_{t,i}$ as regressor's target increment;
  end for
end for

Landmarks-Attention Network

In this section, we will describe the structure of the proposed regressor. Our goal is to learn a mapping between appearance features and landmarks' location increment. Previous works usually first obtain robust initialization location by extracting features in the whole image and then refine the location by many refinement networks (Xiao et al. 2016; Sun, Wang, and Tang 2013) or stack all landmarks patches to directly extract all landmarks features (Trigeorgis et al. 2016). They either require a number of model parameters or generate indiscriminative features. Thus we propose a Landmarks-Attention Network (LAN) to overcome the above two drawbacks. Our regressor is a single CNN which concurrently *pays attention to* appearance feature around each facial landmark. Specifically, for each landmarks patch, we extract features by several convolutional and pooling layers, then concatenate these independent feature vectors and add two fully connected layers to learn a holistic location increment. The structure of each feature extraction sub-network is illustrated in Figure 2(b) and the detailed information of the sub-network is presented in Table 1.

Table 1: Feature extraction sub-network of Landmarks-Attention Network for each patch.

Layer	Input Tensor	Kernel	Output Tensor
conv1	$57 \times 57 \times 3$	$3 \times 3 \times 3 \times 16$	$57 \times 57 \times 16$
pool1	$57 \times 57 \times 16$	2×2	$29 \times 29 \times 16$
conv2	$29 \times 29 \times 16$	$2 \times 2 \times 16 \times 32$	$29 \times 29 \times 32$
pool2	$29 \times 29 \times 32$	2×2	$15 \times 15 \times 32$
conv3	$15 \times 15 \times 32$	$2 \times 2 \times 32 \times 64$	$15 \times 15 \times 64$
pool3	$15 \times 15 \times 64$	2×2	$8 \times 8 \times 64$
fc1	$8 \times 8 \times 64$ ($1 \times 1 \times 4096$)	4096×10	$1 \times 1 \times 10$

Compared to the previous networks, our proposed model has three advantages: (1) The landmarks feature extracted by independent sub-networks can be more discriminative, as showed in Figure 6; (2) Concatenating all independent features vectors and adding fully connected layers can obtain a holistic landmarks location increment, especially when some landmarks are occluded or blurred; (3) Our network is very light, whose parameters number(3.72M in total) is far less than other CNN models (*e.g.*, AlexNet (Krizhevsky, Sutskever, and Hinton 2012) contains about 60M parameters and VGGNet (Simonyan and Zisserman 2014) contains about 138M parameters).

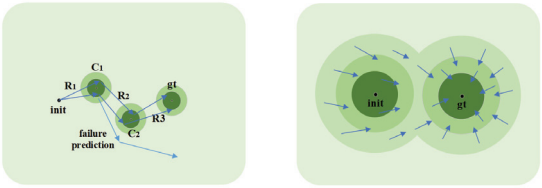
Training

The training process is illustrated in Figure 2 (a) and (b). Since sampling period T can be large enough, online random sampling process can generate nearly unlimited training data $\bigcup_{t=1}^{T}\bigcup_{i=1}^{N}(P_{t,i}, \Delta\theta_{t,i})$. Then, the above described LAN is trained to learn the descent directions in coarse and fine stages together. This process can be formulated as

$$R_\Delta = \underset{R_\Delta}{\operatorname{argmin}} \frac{1}{T \times N} \sum_{t=1}^{T}\sum_{i=1}^{N} \|\Delta\theta_{t,i} - R_\Delta(P_{t,i})\|_2^2, \quad (6)$$

where R_Δ is the target self-iterative regressor (*i.e.*, LAN), and t indicates the t_{th} sampling period.

Since the training space of SIR includes most possible landmarks distribution from coarse stages to fine stages, the training process will generate a Descent Direction Map (DDM) in the sampling space where each sample's descent direction roughly points toward the ground truth. As illustrated in Figure 3 (b), SIR is more robust than CR because the former can cover more training space and isn't affected by the optimization path. When the previous regressor predicts false descent directions, SIR can still converge to the ground truth while CR is prone to drift away.



(a) Cascaded Regression (b) Self-Iterative Regression

Figure 3: (a) Typical cascaded regression process: starting from initial value, parameters are updated and close to the ground truth (such as $init \rightarrow C_1 \rightarrow C_2 \rightarrow gt$) by regressors $R_k(k = 1, 2, 3, ...)$. Once one regressor predicts the false direction, the final result is prone to drift away; (b) SIR Descent Direction Map: the training space of SIR includes distribution from coarse stages to fine stages and all descent directions are pointed to ground truth.

Self-Iterative Updating

For the testing process, similar to the cascaded regression methods, starting from initial location parameters θ_0, we iteratively update the location parameters θ_k and extract new patches P_k till converges. The process is presented in Algorithm 2, and facial landmarks location parameter is updated by,

$$\theta_{k+1} = \theta_k + R_\Delta(P_k), \quad k = 0, 1, \cdots. \quad (7)$$

Algorithm 2 Self-Iterative updating process of SIR

Input: Regressor R_Δ, Initial location θ_0, Total iteration times K

Output: Prediction of facial landmarks' location θ_K

1: **for** $k = 0$ to $K - 1$ **do**
2: Extract patches P_k in location θ_k
3: $\theta_{k+1} = \theta_k + R_\Delta(P_k)$
4: **end for**

Experiments

In this section, we perform experiments to demonstrate the effectiveness of the proposed SIR compared to state-of-the-art methods. Specifically, we evaluate the proposed method model by (1) comparing the performance of SIR vs. state-of-the-art and baseline cascaded regression; (2) comparing the number of model parameters and memory storage of pre-train models; and (3) studying the effect of the proposed feature extraction network(LAN), the number of iteration times and sampling space parameter.

Datasets. The 300-W dataset is short for 300 faces in-the-wild (Sagonas et al. 2016), which is designed for evaluating the performance of facial landmarks detection. The training set ($3,148$ faces in total) consists of AFW dataset (Ramanan 2012), HELEN training set (Le et al. 2012) and LFPW training set (Belhumeur et al. 2011). Two testing sets are established, *i.e.*, *public testing set* (689 faces in total) including HELEN testing set (Le et al. 2012), LFPW testing set (Belhumeur et al. 2011) and IBUG dataset (Sagonas et al. 2016); and *competition testing set* (600 faces in total) including 300 indoor and 300 outdoor faces images.

Metrics. Normalized Mean Error (NME) measures landmarks' mean location error normalized by inter-pupil (eyes centers) distance (Zhu et al. 2015; Ren et al. 2014) or inter-ocular (outer eye corners) distance (Trigeorgis et al. 2016). Cumulative Error Distribution (CED) curve is the cumulative distribution function of the normalized error, which can avoid heavily impacted by some big failures (Yang et al. 2015). We also calculated another two evaluation metrics, namely Area-Under-the-Curve (AUC_α) and Failure Rate (FR_α). Similar as MDM (Trigeorgis et al. 2016), we consider mean point-to-point error greater than 0.08 as a failure, *i.e.*, $\alpha = 0.08$.

Implementation Detail. We perform the experiments based on a machine with Core i7-5930k CPU, 32 GB memory and GTX 1080 GPU with 8G video memory. The detected faces are resized into 256×256 and the location patch size is 57×57. For CNN structure, the Rectified Linear Unit (ReLU) is adopted as the activation function, and the optimizer is the Adadelta (Zeiler 2012) approach, learning rate is set to 0.1 and weight decay is set to $1e-4$. Training the CNN requires around 2 days.

Comparison with State-of-the-arts

As shown in Table 2, we compare the proposed method with several state-of-the-art facial landmarks detection methods in the public testing set. Specifically, the *common subset* consists of LFPW testing set (224 faces) and HELEN testing set (330 faces) and the *challenging subset* is IBUG dataset (135 faces). Thus the the *full set* (689 faces) of the union of the common (554 faces) and challenging subsets (135 faces). The NME results shows that SIR performs comparatively with RAR (Xiao et al. 2016) and outperform other existing methods (Cao et al. 2014; Burgos-Artizzu, Perona, and Dollár 2013; Xiong and la Torre 2013; Ren et al. 2014; Zhu et al. 2015; Kowalski and Naruniec 2016; Trigeorgis et al. 2016; Xiao et al. 2016). Besides, more visual results are also illustrated in Figure 9. In the more challenging IBUG subset, our method achieves robust performance in large pose, expression and illumination environment.

Table 2: NME (inter-pupil normalization) results in the public testing set. The top two performance are shown in boldface.

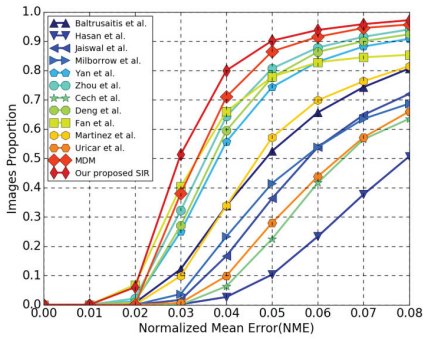
Methods	Common subset	Challenging subset	Full set
RCPR(2013)	6.18	17.26	8.35
ESR(2012)	5.28	17.00	7.58
SDM(2013)	5.57	15.40	7.50
LBF(2014)	4.95	11.98	6.32
CFAN(2014)	5.55	-	-
CFSS(2015)	4.73	9.98	5.76
Kowalski *et al.*(2016)	4.62	9.48	5.57
MDM(2016)	4.83	10.14	5.88
RAR(2016)	**4.12**	**8.35**	**4.94**
SIR	**4.29**	**8.14**	**5.04**

On the other hand, we evaluate SIR in the competition testing set. As shown in Figure 4, the SIR method outperform the state-of-the-art methods (Cech et al. 2016; Deng et al. 2016; Fan and Zhou 2016; Baltrusaitis, Robinson, and Morency 2013; Yan et al. 2013; Zhou et al. 2013; Uricár et al. 2016; Jaiswal, Almaev, and Valstar 2013; Milborrow, Bishop, and Nicolls 2013; Kamrul Hasan, Pal, and Moalem 2013; Martinez and Valstar 2016) according to the CED curve. Moreover, Table 3 presents the quantitative results for both the 51-point and 68-point error metrics (*i.e.*, AUC and Failure Rate at a threshold of 0.08 of the normalised error), compared to existing methods (Kazemi and Sullivan 2014; Tzimiropoulos 2015; Asthana et al. 2014; Xiong and la Torre 2013; Zhou et al. 2013; Yan et al. 2013; Uricár et al. 2016; Zhu et al. 2015; Trigeorgis et al. 2016). The promising performances on two metrics indicate the effectiveness of the proposed method.

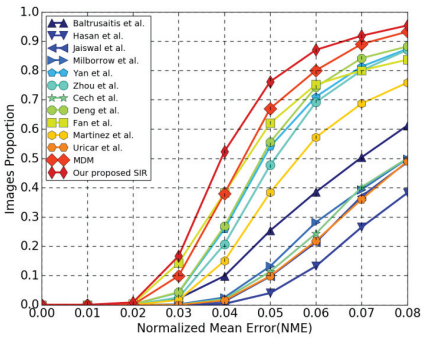
Comparison with Cascaded Regression

As discussed before, previous cascaded regression methods adding more regressors can achieve better performance, but increase the number of model parameters, computing resources and storage space, especially for deep learning based methods. Different from them, our method obtains state-of-the-art performance by iterative call the same regressor rather than adding any more regressors.

Our method reduces the model complexity while keeps the performance in two folds: (1) the proposed network focuses on the landmarks' local feature, which significantly reduces the dimension of final feature layer; (2) only one CNN



(a) CED curve of facial *51-points*.



(b) CED curve of facial *68-points*.

Figure 4: CED curve results comparison in 300-W competition testing set.

Table 3: Quantitative results using $AUC_{0.08}(\%)$ and $FR_{0.08}(\%)$ in the competition testing set. *51-points* and *68-points* are two groups of facial landmarks and *51-points* is part of *68-points*.

	51-points		**68-points**	
Method	AUC	Failure	AUC	Failure
ERT(2014)	40.60	13.50	32.35	17.00
PO-CR(2015)	47.65	11.70	-	-
Chehra(2014)	31.12	39.30	-	-
Intraface(2013)	38.47	19.70	-	-
Balt *et al.*(2013)	37.65	17.17	19.55	38.83
zhou *et al.*(2013)	53.29	5.33	32.81	13.00
Yan *et al.*(2013)	49.07	8.33	34.97	12.67
CFSS(2015)	50.79	7.80	39.81	12.30
MDM(2016)	56.34	4.20	45.32	6.80
SIR	**58.11**	**2.83**	**46.56**	**4.33**

module is required to iteratively predict the location parameters, while cascaded regression usually requires at least three regressors(Trigeorgis et al. 2016; Xiong and la Torre 2013).

To prove the effectiveness of SIR, we add a baseline CR method which extracts features by the same LAN while adopts cascaded regression framework. Both baseline CR and SIR is updated for 4 times before the stable performance. As shown in Table 4, our method requires parameters and memory far less than other cascaded regression based methods.

Table 4: Comparison with state-of-the-art methods in the public testing set, with the first and the second best results highlighted. DL indicates whether the method is based on deep learning.

Method	DL	NME	# params	model memory
RCPR(2013)	N	8.35	-	91.3MB
LBF(2014)	N	6.32	-	36.6MB
CFSS(2015)	N	5.76	-	225.2MB
MDM(2016)	Y	5.61	80.56M	322.3MB
RAR(2016)	Y	**4.94**	15.65M+	62.6MB+
baseline CR	Y	6.23	**14.88M**	**62.4MB**
SIR	Y	**5.04**	**3.72M**	**15.6MB**

Discussion and Analysis

In this section, we perform analyses on the effect of several important modules in our method to the final performance.

Effect of different feature extraction networks. In SIR framework, we adopt the Landmarks-Attention Network (we call it SIR-LAN) to extract landmarks patches features separately, while some works stack all landmarks patches and then extract the whole features directly (we call it SIR-Stack), as illustrated in Figure 5. To demonstrate the effectiveness of our network, we conduct an experiment by SIR framework to compare the above two networks with the same number of CNN layers and model parameters, the structure of SIR-Stack is showed in Figure 5. The result illustrated in Figure 6 shows that the proposed network extracting patches features separately performs significantly better than previous methods extracting patches feature together (*e.g.*, MDM (Trigeorgis et al. 2016)).

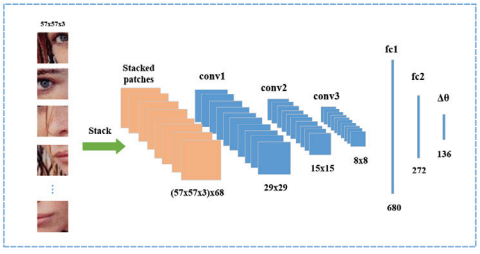


Figure 5: Structure of SIR-Stack Network. For a fair comparison, we adopt the same number of CNN layers and model parameters(3.72M).

Effect of iteration times. From Figure 7, we can find that the accuracy will be improved by adding iteration times before the stable performance (*i.e.*, 4 iterations) is achieved. When increasing iteration times, more model memory will be added in baseline CR.

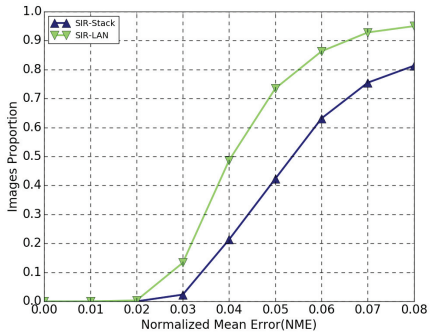


Figure 6: Comparison between SIR-LAN and SIR-Stack in the competition testing set.

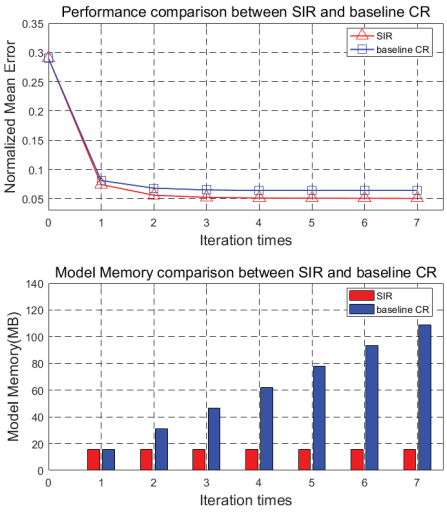


Figure 7: Effect of iteration times. Top: Comparison between SIR and baseline CR in accuracy. With the increase of iteration times, both SIR and baseline CR can decrease the detection error and SIR performs better than baseline CR. Bottom: Comparison between SIR and baseline CR in in Model Memory. Increasing the iteration times will increase its model memory of baseline CR, while SIR doesn't because it can iteratively call itself.

Effect of Gaussian sampling space parameters. As one of the most important processes, random sampling space significantly affects the final robustness and accuracy. As shown in Figure 8, the NME results are presented by varying the standard deviation σ of Gaussian sampling. Appropriate values lead to promising performance so that we set $\sigma = 0.2$ in our method.

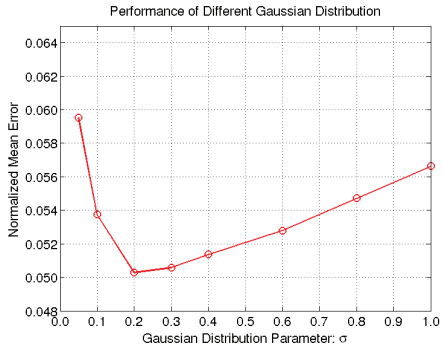


Figure 8: Performances of different Gaussian sampling in the 300-W public testing set.

Figure 9: Several facial landmarks detection results in 300-W public testing and competition testing set. Blue dot in each sub-picture indicates ground truth landmarks location and yellow dot indicates the predicted location of SIR. Pictures for the five rows are from HELEN testing set, LFPW testing set, IBUG set, 300-W competition testing Indoor and Outdoor set respectively.

Conclusion

In this paper, we develop a SIR framework solve the non-linear least squares problems. Compared with cascaded regression, it only needs to train a single regressor to learn descent directions in coarse stages to fine stages together, and refines the target parameters iteratively by call the same regressor. Experimental results in the facial landmarks detection task demonstrate that the proposed self-iterative regressor achieves comparable accuracy to state-of-the-art methods, but significantly reduces the number of parameters and memory storage of the pre-trained models. In the future, we will extend the proposed method to other applications, such as human pose prediction, structure from motion and 3D face reconstruction.

References

Asthana, A.; Zafeiriou, S.; Cheng, S.; and Pantic, M. 2014. Incremental face alignment in the wild. In *CVPR*, 1859–1866.

Baltrusaitis, T.; Robinson, P.; and Morency, L.-P. 2013. Constrained local neural fields for robust facial landmark detection in the wild. In *ICCVW*, 354–361.

Belhumeur, P. N.; Jacobs, D. W.; Kriegman, D. J.; and Kumar, N. 2011. Localizing parts of faces using a consensus of exemplars. In *CVPR*, 545–552.

Blanz, V., and Vetter, T. 1999. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, 187–194.

Breiman, L. 2001. Random forests. *Machine Learning* 45(1):5–32.

Burgos-Artizzu, X. P.; Perona, P.; and Dollár, P. 2013. Robust face landmark estimation under occlusion. In *ICCV*, 1513–1520.

Cao, X.; Wei, Y.; Wen, F.; and Sun, J. 2014. Face alignment by explicit shape regression. *IJCV* 107(2):177–190.

Carreira, J.; Agrawal, P.; Fragkiadaki, K.; and Malik, J. 2016. Human pose estimation with iterative error feedback. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 4733–4742.

Cech, J.; Franc, V.; Uricár, M.; and Matas, J. 2016. Multi-view facial landmark detection by using a 3d shape model. *IVC* 47:60–70.

Deng, J.; Liu, Q.; Yang, J.; and Tao, D. 2016. M3 csr. *IVC* 47(C):19–26.

Dollár, P.; Welinder, P.; and Perona, P. 2010. Cascaded pose regression. In *CVPR*, 1078–1085.

Fan, H., and Zhou, E. 2016. Approaching human level facial landmark localization by deep learning. *IVC* 47(C):27–35.

Jaiswal, S.; Almaev, T. R.; and Valstar, M. F. 2013. Guided unsupervised learning of mode specific models for facial point detection in the wild. In *ICCV*.

Kamrul Hasan, M.; Pal, C.; and Moalem, S. 2013. Localizing facial keypoints with global descriptor search, neighbour alignment and locally linear models. In *ICCV*.

Kazemi, V., and Sullivan, J. 2014. One millisecond face alignment with an ensemble of regression trees. In *CVPR*, 1867–1874.

Kowalski, M., and Naruniec, J. 2016. Face alignment using k-cluster regression forests with weighted splitting. *SPL* 23(11):1567–1571.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, 1106–1114.

Le, V.; Brandt, J.; Lin, Z.; Bourdev, L.; and Huang, T. S. 2012. Interactive facial feature localization. In *ECCV*, 679–692.

Liu, F.; Zeng, D.; Zhao, Q.; and Liu, X. 2016. Joint face alignment and 3d face reconstruction. In *ECCV*, 545–560.

Lowe, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2):91–110.

Martinez, B., and Valstar, M. F. 2016. L2,1-based regression and prediction accumulation across views for robust facial landmark detection. *IVC* 47:36–44.

Milborrow, S.; Bishop, T. E.; and Nicolls, F. 2013. Multiview active shape models with sift descriptors for the 300-w face landmark challenge. In *ICCV*.

Ramanan, D. 2012. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2879–2886.

Ren, S.; Cao, X.; Wei, Y.; and Sun, J. 2014. Face alignment at 3000 FPS via regressing local binary features. In *CVPR*, 1685–1692.

Sagonas, C.; Antonakos, E.; Tzimiropoulos, G.; Zafeiriou, S.; and Pantic, M. 2016. 300 faces in-the-wild challenge. *IVC* 47(C):3–18.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556.

Sun, Y.; Wang, X.; and Tang, X. 2013. Deep convolutional network cascade for facial point detection. In *CVPR*, 3476–3483.

Trigeorgis, G.; Snape, P.; Nicolaou, M. A.; Antonakos, E.; and Zafeiriou, S. 2016. Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In *CVPR*, 4177–4187.

Tu, Z. 2008. Auto-context and its application to high-level vision tasks. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*.

Tzimiropoulos, G. 2015. Project-out cascaded regression with an application to face alignment. In *CVPR*, 3659–3667.

Uricár, M.; Franc, V.; Thomas, D.; Sugimoto, A.; and Hlavác, V. 2016. Multi-view facial landmark detector learned by the structured output SVM. *IVC* 47:45–59.

Xiao, S.; Feng, J.; Xing, J.; Lai, H.; Yan, S.; and Kassim, A. A. 2016. Robust facial landmark detection via recurrent attentive-refinement networks. In *ECCV*, 57–72.

Xiong, X., and la Torre, F. D. 2013. Supervised descent method and its applications to face alignment. In *CVPR*, 532–539.

Xiong, X., and la Torre, F. D. 2015. Global supervised descent method. In *CVPR*, 2664–2673.

Yan, J.; Lei, Z.; Yi, D.; and Li, S. Z. 2013. Learn to combine multiple hypotheses for accurate face alignment. In *ICCVW*, 392–396.

Yang, H.; Jia, X.; Loy, C. C.; and Robinson, P. 2015. An empirical study of recent face alignment methods. *CoRR* abs/1511.05049.

Zeiler, M. D. 2012. ADADELTA: an adaptive learning rate method. *CoRR* abs/1212.5701.

Zhang, J.; Shan, S.; Kan, M.; and Chen, X. 2014a. Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment. In *ECCV*, 1–16.

Zhang, Z.; Luo, P.; Loy, C. C.; and Tang, X. 2014b. Facial landmark detection by deep multi-task learning. In *ECCV*, 94–108.

Zhou, E.; Fan, H.; Cao, Z.; Jiang, Y.; and Yin, Q. 2013. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *ICCVW*, 386–391.

Zhu, S.; Li, C.; Loy, C. C.; and Tang, X. 2015. Face alignment by coarse-to-fine shape searching. In *CVPR*, 4998–5006.

Zhu, X.; Lei, Z.; Liu, X.; Shi, H.; and Li, S. Z. 2016. Face alignment across large poses: A 3d solution. In *CVPR*, 146–155.