2017 IEEE 12th International Conference on Automatic Face & Gesture Recognition

A Fully End-to-End Cascaded CNN for Facial Landmark Detection

Zhenliang He[1,2] Meina Kan[1,3] Jie Zhang[1,2] Xilin Chen[1] Shiguang Shan[1,3]
[1] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] CAS Center for Excellence in Brain Science and Intelligence Technology

Abstract— **Facial landmark detection plays an important role in computer vision. It is a challenging problem due to various poses, exaggerated expressions and partial occlusions. In this work, we propose a Fully End-to-End Cascaded Convolutional Neural Network (FEC-CNN) for more promising facial landmark detection. Specifically, FEC-CNN includes several sub-CNNs, which progressively refine the shape prediction via finer and finer modeling, and the overall network is optimized fully end-to-end. Experiments on three challenging datasets, IBUG, 300W competition and AFLW, demonstrate that the proposed method is robust to large poses, exaggerated expressions and partial occlusions. The proposed FEC-CNN significantly improves the accuracy of landmark prediction.**

I. INTRODUCTION

Facial landmark detection is a vital topic in computer vision, which is widely used in face recognition, face animation and facial expression recognition. In the past few decades, many researchers devote to tackle this problem and impressive progress has been achieved.

Among early works, the Active Shape Model (ASM) [6][8] and the Active Appearance Model (AAM) [5][19] are representative ones, which both use Principal Component Analysis (PCA) to parameterize shape and appearance to tackle facial landmark detection problem. They are however hard to model complex variations and thus perform badly on wild testing data, partially because PCA is a linear model which can hardly characterize those nonlinear variations.

Later, regression based methods, especially cascaded regression methods achieve great success on facial landmark detection [7][4][26][3][20][28]. Cascaded Pose Regression (CPR) [7] firstly proposes cascaded regression framework for general object pose estimation, which designs shape-indexed control point feature and random fern regressors to predict shapes of faces, mice and fish. Another representative approach in cascaded regression method is Supervised Descent Method (SDM) [26], which cascades several linear regressors to predict the shape stage by stage by using shape-indexed SIFT feature [18]. SDM achieves quite promising results on LFPW [2] and RU-FACS [1] datasets for real-time facial landmark detection and tracking. Inspired by the promising performance of SDM, the regression based methods draw more and more attentions. In [3], Burgos-Artizzu et al. further propose Robust Cascaded Pose Regression (RCPR) to explicitly handle occlusion problems, which simultaneously predicts the locations and occlusion conditions of each landmark. Moreover, the interpolated shape-indexed feature and smart restart strategy are exploited to improve the robustness against large pose variations and tackle shape initialization problem respectively. Instead of directly using random forest or random fern for face shape regression like CPR and RCPR, Ren et al. [20] propose to learn local binary feature based on random forest and predict the landmarks by linear regression, achieving both high accuracy and efficiency. The aforementioned methods all need an initial face shape to start with. So, the optimization of landmark prediction might fail into local minimum under large pose variations when given a poor initialization. To further improve the robustness to large pose variations, Zhu et al. [28] propose Coarse-to-Fine Shape Searching (CFSS). Rather than give an initial shape, CFSS uses a cascaded shape searching method to relieve local optima problem due to poor shape initialization. Although cascaded methods achieve promising performance on both controlled and wild settings, most of them are linear or shallow models which might be insufficient for modeling the complex nonlinearity from the face appearance to shape, especially under challenging wild scenario.

In consideration of the strong capacity of deep model in nonlinear modeling, it is naturally applied to facial landmark detection and achieved further huge success [24][27][25]. Sun et al. [24] propose to cascade several deep convolutional neural networks (DCNN) to predict the shape stage by stage, in which the feature is optimized by using the DCNN rather than hand-crafting. Furthermore, Zhang et al. [27] propose a Coarse-to-Fine Auto-Encoder Network (CFAN) which cascades several stacked auto-encoder in a coarse-to-fine way to refine the shape stage by stage. CFAN achieves promising performance, however CFAN employs hand-crafted SIFT feature as input. Except for the above 2D methods, a few 3D methods combining with deep cascaded model also make promising progress for unconstrained facial landmark detection [12][13][30]. In [13], Jourabloo et al. design a cascaded CNN regression framework integrating a dense 3D Morphable Model that predicts 3D shape parameters and projects 3D landmark coordinates to 2D.

In all above-mentioned cascaded methods, the models of all stages are optimized one by one, which means each stage heavily depends on the previous one. Consequently, the learnt model might be suboptimal, especially when the results from the previous stage are bad. To tackle this problem, Mnemonic

978-1-5090-4023-0/17 $31.00 © 2017 IEEE
DOI 10.1109/FG.2017.33

IEEE computer society

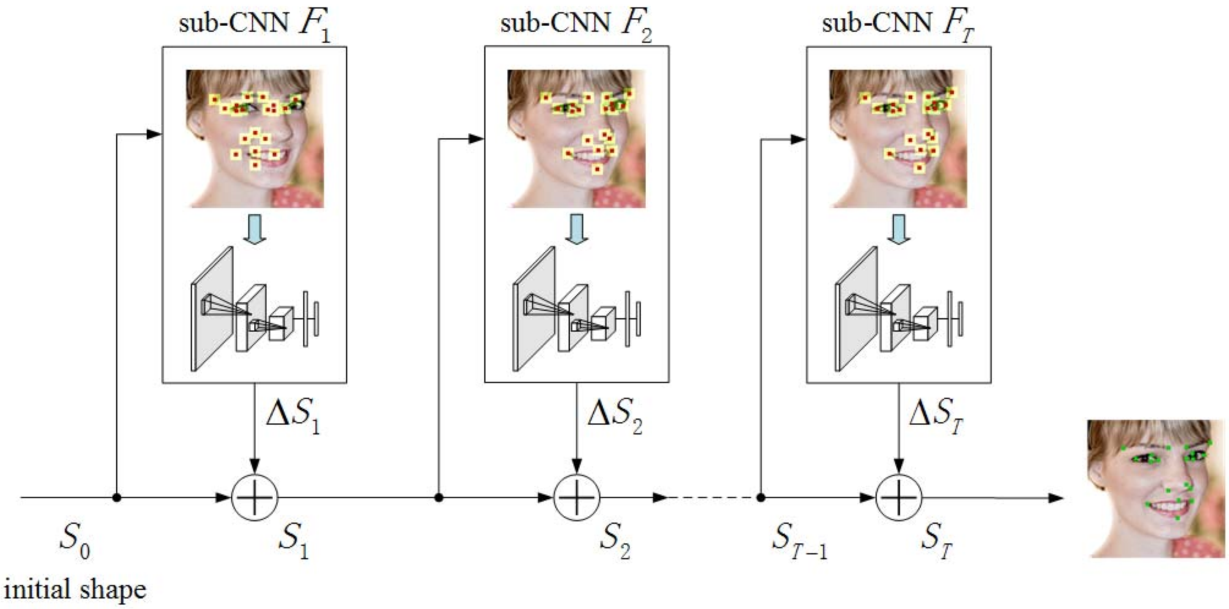


Fig. 1. Overview of the fully end-to-end cascaded convolutional neural network (FEC-CNN) for facial landmark detection. FEC-CNN contains several sub-CNNs each of which takes patches around landmarks predicted by previous stage as input. S_0 denotes the initial shape, $\Delta S_t, t = 1, \cdots, T$ denotes the t^{th} shape residual predicted by the t^{th} sub-CNN and $S_t, t = 1, \cdots, T$ denotes the t^{th} refined shape which is the sum of S_{t-1} and ΔS_t. The whole network is learnt fully end-to-end by propagating down the gradient back through every arrow shown in the figure.

Descent Method (MDM) [25] proposes a recurrent network which enables end-to-end training of a cascaded structure. Similar to DCNN [24], at each stage, MDM takes patches around landmarks predicted by previous stage as input. But differently, with the help of the recurrent structure, each stage of MDM also takes the hidden-layer features of previous stage as input. Therefore, the gradient can be propagated from the last stage to the first one through the connection between the hidden-layers of adjacent stages. That is, MDM allows end-to-end training of the whole network rather than train all the stages separately.

However, MDM is not fully end-to-end as the last layer of each stage is not included in the end-to-end learning, but only the hidden layers are involved, although the gradient is indeed back propagated through all stages. In other words, MDM only considers the gradient from a stage to the hidden layers of previous stage, but does not back propagate the gradient from a stage to the shape prediction layer of its previous stage. Nevertheless, a current stage is evidently related to the output shape of its previous stage, so we argue that directly linking the first layer of current stage and shape prediction from the last layer of previous stage in an explicit end-to-end scheme can benefit the facial landmark detection.

Based on the above arguments, we design a Fully End-to-End Cascaded Convolutional Neural Network (FEC-CNN) as shown in Fig. 1. The proposed FEC-CNN consists of several sub-CNNs, each directly taking the input image and the output shape of its previous stage as input. As a result, the gradient can be back propagated from the loss layer to all the units of each stage, leading to promising performance for facial landmark detection even on wild data.

The rest of this paper is organized as follows: Section II illustrates the details about formulation and optimization of the proposed method. Section III gives the experimental results and analysis on three challenging datasets. Section IV makes conclusions of this work.

II. Our Approach

A. Formulation

Facial landmark detection can be formulated as learning a complex nonlinear mapping from face image I to shape S. Inspired by the powerful ability of deep neural network for modelling nonlinearity, we propose a novel fully end-to-end convolutional neural network as shown in Fig. 1. The proposed method consists of several sub-CNNs, denoted as $\{F_t\}_{t=1}^{T}$, and the whole network is denoted as H. The objective of the proposed method is to learn an end-to-end network H which can well characterize the nonlinear mapping from the appearance to shape as follows:

$$S = H(I) = \sum_{t=1}^{T} \Delta S_t + S_0 \tag{1}$$

$$\Delta S_t = F_t(\Theta(I, S_{t-1})), t = 1, \cdots, T \tag{2}$$

$$S_t = S_{t-1} + \Delta S_t, t = 1, \cdots, T \tag{3}$$

where F_t denotes the sub-CNN in the t^{th} stage, S_0 denotes the initial shape such as mean shape, ΔS_t denotes the shape residual predicted by the t^{th} sub-CNN, S_t denotes the t^{th} refined shape outputted from the t^{th} sub-CNN and $\Theta(I, S)$ is

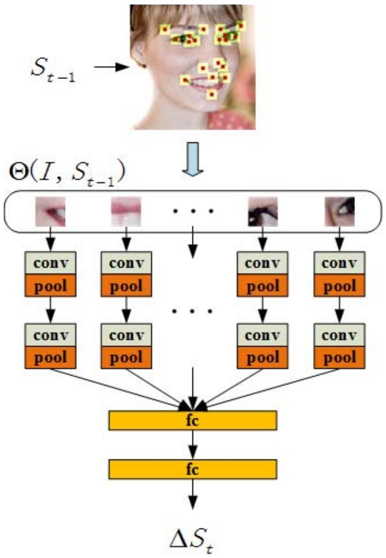


Fig. 2. An illustration of each sub-CNN F_t. At t^{th} stage, shape-indexed patches are extracted from the original input image around the landmarks S_{t-1} predicted by the previous network. Each patch is separately fed into the convolutional layers, then the convolutional features of all patches are concatenated together and go through the fully-connected layers to predict the shape residual ΔS_t.

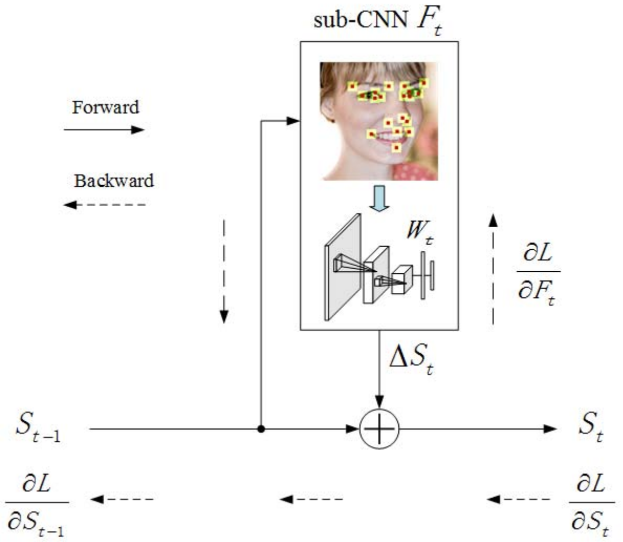

sub-CNN F_t
Forward
Backward
W_t
$\frac{\partial L}{\partial F_t}$
ΔS_t
S_{t-1}
S_t
$\frac{\partial L}{\partial S_{t-1}}$
$\frac{\partial L}{\partial S_t}$

Fig. 3. An illustration of the back propagation process of the sub-CNN in the t^{th} stage. $\frac{\partial L}{\partial S_t}$ denotes the shape gradient coming from the $(t+1)^{th}$ stage, $\frac{\partial L}{\partial F_t}$ denotes the CNN gradient of the t^{th} stage, W_t denotes the parameters of the CNN, and $\frac{\partial L}{\partial S_{t-1}}$ denotes the shape gradient which is propagated to the $(t-1)^{th}$ stage.

the patch extracting function which extracts patches around shape S.

As seen from (2), in the t^{th} stage, by taking the shape-indexed patches extracted around the output shape of the previous $(t-1)^{th}$ stage as input, the sub-CNN F_t attempts to further predict the shape residual ΔS_t between S_{t-1} and the ground truth shape. In other words, the output from one stage is directly used as input of the successive stage, so the stages are strongly related to each other. Furthermore, by summing up the shape prediction of all stages together in (1), all stages are connected under a single objective to approach the ground truth shape. Therefore, all stages can be optimized jointly in an end-to-end way. Besides, each sub-CNN taking the shape-indexed patches from the previous stage as input refines the shape more and more accurately.

Specifically, each sub-CNN F_t in (2) endeavors to further refine the output shape of its previous stage by taking the shape-indexed patches as input to approximate the residual between the output shape of the previous stage and the ground truth shape. As shown in Fig. 2, at each stage, the shape-indexed patches are extracted from the original input image around the landmarks predicted by the previous network. Then, each patch is separately fed into the convolutional layers, and then the convolutional features of all patches are concatenated and followed by fully connected layers to predict the shape residual.

By summing up all stages together, the overall objective of the whole network is formulated as follow:

$$\{F_t^*\}_{t=1}^T = \underset{\{F_t\}_{t=1}^T}{argmin} \sum_{i=1}^N ||\hat{S}^i - \sum_{t=1}^T F_t(\Theta(I^i, S_{t-1}^i)) - S_0^i||_2^2 \quad (4)$$

where $\hat{S}^i$ is the ground truth shape of the i^{th} training sample I^i, and S_t^i is the output shape of the t^{th} sub-CNN for the i^{th} training sample I^i calculated as

$$S_t^i = S_{t-1}^i + F_t(\Theta(I^i, S_{t-1}^i)). \quad (5)$$

The $\{F_t\}_{t=1}^T$ can be obtained jointly by optimizing the whole network end-to-end detailed in the following.

B. Optimization

Following most existing deep neural network, the objective in (4) is optimized by using the gradient descent method. Firstly, the gradient of the parameters is calculated, and then the parameters are updated along the descent direction of the gradient.

As seen from Fig. 1, the structures of all stages are similar, and so the gradient back propagation processes of all stages are also similar. Let's take the t^{th} stage shown in Fig. 3 as an example.

Let $L_i = ||\hat{S}^i - \sum_{t=1}^T F_t(\Theta(I^i, S_{t-1}^i)) - S_0^i||_2^2$ denote the prediction loss of the i^{th} training sample, and $\Delta S_t^i = F_t(\Theta(I^i, S_{t-1}^i))$. We omit the index i of the i^{th} sample for simplicity unless misunderstanding. Let W_t denote the parameters of CNN F_t.

As seen from Fig. 3, in the t^{th} stage, given the gradient $\frac{\partial L}{\partial S_t}$, there are two parts of gradient needed to be calculated: 1) the gradient of the parameters W_t of the sub-CNN F_t which is further used to update the sub-CNN; 2) the gradient of the input shape S_{t-1} which is propagated to the previous $(t-1)^{th}$ stage.

Firstly, the gradient of W_t is calculated as follows:

$$\begin{aligned}\frac{\partial L}{\partial W_t} &= \frac{\partial L}{\partial S_t}\frac{\partial S_t}{\partial \Delta S_t}\frac{\partial \Delta S_t}{\partial W_t} \\ &= \frac{\partial L}{\partial S_t}\frac{\partial (\Delta S_t + S_{t-1})}{\partial \Delta S_t}\frac{\partial \Delta S_t}{\partial W_t} \\ &= \frac{\partial L}{\partial S_t}\frac{\partial \Delta S_t}{\partial W_t}, t=1,\cdots,T\end{aligned} \tag{6}$$

In (6), $\frac{\partial L}{\partial S_t}$ is calculated from the $(t+1)^{th}$ stage, and $\frac{\partial \Delta S_t}{\partial W_t}$ can be easily obtained like any typical CNN.

Secondly, the gradient of the input shape $\frac{\partial L}{\partial S_{t-1}}$ is calculated as below:

$$\begin{aligned}\frac{\partial L}{\partial S_{t-1}} &= \frac{\partial L}{\partial S_t}\frac{\partial S_t}{\partial S_{t-1}} \\ &= \frac{\partial L}{\partial S_t}\frac{\partial (\Delta S_t + S_{t-1})}{\partial S_{t-1}} \\ &= \frac{\partial L}{\partial S_t}\left(1+\frac{\partial \Delta S_t}{\partial \Theta_t}\frac{\partial \Theta_t}{\partial S_{t-1}}\right), t=1,\cdots,T\end{aligned} \tag{7}$$

where $\Theta_t = \Theta(I, S_{t-1})$. In (7), $\frac{\partial L}{\partial S_t}$ is calculated from the $(t+1)^{th}$ stage , and $\frac{\partial \Delta S_t}{\partial \Theta_t}$ can be easily obtained like most typical CNN. So, the left part is how to calculate the gradient of S_{t-1} from the shape-indexed patch, i.e. $\frac{\partial \Theta_t}{\partial S_{t-1}}$, which will be detailed in the following.

With (6) and (7), the gradient can be propagated stage by stage by looping t from T down to 1. For the start, the gradient of the output shape of the last stage is computed as

$$\frac{\partial L}{\partial S_T} = 2(S_T - \hat{S}). \tag{8}$$

In summary, with (8) as start, the whole network in (4) can be updated end-to-end by looping t according to (6) and (7). The above process is for one sample, when given a batch of a few samples, the gradient of each sample can be obtained similarly and independently.

Now, let's concentrate on $\frac{\partial \Theta_t}{\partial S_{t-1}}$ which denotes the derivative of shape-indexed patches with respect to the input shape. As the shape-indexed patches are extracted independently, the $\frac{\partial \Theta_t}{\partial S_{t-1}}$ can be divided into a few independent $\frac{\partial \Theta_{tk}}{\partial S_{(t-1)k}}$, with Θ_{tk} denoting the patch extracted around the k^{th} landmark $S_{(t-1)k} = (y_{(t-1)k}, x_{(t-1)k})$. For simplicity, the index k and t is omitted unless misunderstanding.

Formally, the patch extraction process Θ around one landmark (y,x) can be formulated as below:

$$\Theta : I, y, x \rightarrow V \tag{9}$$

where I is the input image with height H and width W, (y,x) is the coordinate of a landmark, and V is the output patch with height h and width w.

Therefore, the $\frac{\partial \Theta_t}{\partial S_{t-1}}$ can be divided into multiple independent gradient w.r.t. each patch, i.e. $\frac{\partial V}{\partial x}$ and $\frac{\partial V}{\partial y}$. Inspired by Spatial Transformer Networks [11], we adopt bilinear interpolation for generating patch V. The pixel located at (q,p) of V is calculated via the following function,

$$V_{qp} = \sum_{n=0}^{H-1}\sum_{m=0}^{W-1} I_{nm} max(0, 1-|y_q - n|) max(0, 1-|x_p - m|) \tag{10}$$

$$y_q = y + q - (h-1)/2 \tag{11}$$

$$x_p = x + p - (w-1)/2 \tag{12}$$

where (y_q, x_p) is the coordinate of V_{qp} with respect to the whole image I, h and w denote the height and the width of the patch V, I_{nm} denotes the pixel located at (n,m) of I.

The partial derivative of (10) w.r.t. the landmark is

$$\frac{\partial V_{qp}}{\partial x} = \frac{\partial V_{qp}}{\partial x_p}\frac{\partial x_p}{\partial x} = \frac{\partial V_{qp}}{\partial x_p} \tag{13}$$

$$\frac{\partial V_{qp}}{\partial x_p} = \sum_{n=0}^{H-1}\sum_{m=0}^{W-1} I_{nm} max(0, 1-|y_q - n|) \begin{cases} 0, & |m - x_p| \geq 1; \\ 1, & m \geq x_p; \\ -1, & m < x_p \end{cases} \tag{14}$$

Similarly, $\frac{\partial V_{qp}}{\partial y}$ is calculated as below:

$$\frac{\partial V_{qp}}{\partial y} = \frac{\partial V_{qp}}{\partial y_q}\frac{\partial y_q}{\partial y} = \frac{\partial V_{qp}}{\partial y_q} \tag{15}$$

$$\frac{\partial V_{qp}}{\partial y_q} = \sum_{n=0}^{H-1}\sum_{m=0}^{W-1} I_{nm} max(0, 1-|x_p - m|) \begin{cases} 0, & |n - y_q| \geq 1; \\ 1, & n \geq y_q; \\ -1, & n < y_q \end{cases} \tag{16}$$

The time cost of (10), (14) and (16) are $O(WH)$ which is time consuming. However, we found that the time cost can be reduced to $O(1)$, because V_{qp} can be obtained by only considering four neighboring pixels of (y_q, x_p) in I. Specifically, (10), (14) and (16) can be equally reformulated as follows,

$$V_{qp} = I_{\lfloor y_q \rfloor \lfloor x_p \rfloor} y_d x_d + I_{\lfloor y_q \rfloor \lceil x_p \rceil} y_d x_u + I_{\lceil y_q \rceil \lfloor x_p \rfloor} y_u x_d + I_{\lceil y_q \rceil \lceil x_p \rceil} y_u x_u \tag{17}$$

$$\frac{\partial V_{qp}}{\partial x_p} = -I_{\lfloor y_q \rfloor \lfloor x_p \rfloor} y_d + I_{\lfloor y_q \rfloor \lceil x_p \rceil} y_d - I_{\lceil y_q \rceil \lfloor x_p \rfloor} y_u + I_{\lceil y_q \rceil \lceil x_p \rceil} y_u \tag{18}$$

$$\frac{\partial V_{qp}}{\partial y_q} = -I_{\lfloor y_q \rfloor \lfloor x_p \rfloor} x_d - I_{\lfloor y_q \rfloor \lceil x_p \rceil} x_u + I_{\lceil y_q \rceil \lfloor x_p \rfloor} x_d + I_{\lceil y_q \rceil \lceil x_p \rceil} x_u \tag{19}$$

$$x_d = 1 - (x_p - \lfloor x_p \rfloor) \tag{20}$$

$$x_u = 1 - (\lceil x_p \rceil - x_p) \tag{21}$$

$$y_d = 1 - (y_q - \lfloor y_q \rfloor) \tag{22}$$

$$y_u = 1 - (\lceil y_q \rceil - y_q) \tag{23}$$

Following (13), (18), (15) and (19), each independent part of $\frac{\partial \Theta_t}{\partial S_{t-1}}$, i.e. $\frac{\partial \Theta_{tk}}{\partial S_{(t-1)k}}$ can be calculated inducing the $\frac{\partial L}{\partial S_{t-1}}$ in (7).

C. Discussions

Differences with DCNN [24]. Our FEC-CNN and DCNN differ in the following aspects: 1) DCNN optimizes different stages separately, while FEC-CNN optimizes all stages fully end-to-end which results in a better solution. 2) DCNN predicts different groups of landmarks separately without any shape constraint, which might be stuck in partial occlusions. On the contrary, in FEC-CNN, the CNN features of all landmarks are concatenated together as input of the fully connected layer in each stage which implicitly constrain all landmarks in a reasonable shape even with partial occlusions.

Differences with CFAN [27]. FEC-CNN and CFAN differ in the following aspects: 1) Similar as DCNN, CFAN is optimized stage by stage, while our FEC-CNN is optimized fully end-to-end. 2) CFAN employs SIFT feature [18] as shape-indexed feature which is handcrafted and not differentiable. On the contrary, out FEC-CNN employs CNN to directly learn the feature based on the shape-indexed patches which is more flexible and differentiable.

Differences with MDM [25]. Both MDM and FEC-CNN are structured end-to-end. However, MDM only considers the gradient from a stage to the hidden layers of the previous stage, but doesn't consider the gradient from a stage to the shape prediction layer of the previous stage, therefore, MDM is not fully end-to-end. On the contrary, in FEC-CNN, the gradient is back propagated from the loss layer to all units of each stage, which is fully end-to-end.

III. Experiments

A. Experimental Settings

Datasets. To evaluate the proposed method, we employ three wild datasets, i.e. 300W [23][22], 300W competition [21][22] and AFLW [15] which are commonly used for facial landmark detection.

The 300W dataset consists of three wild datasets including LFPW [2], AFW [31], HELEN [17] and a challenging dataset IBUG [22]. The 300W competition dataset consists of an indoor subset and an outdoor subset. The 68-point annotations of these datasets are provided by Sagonas et al. [23][21][22]. The AFLW dataset contains 24386 in-the-wild faces with large variation of head pose which is challenging for multi-view facial landmark detection.

The 300W dataset and the 300W competition dataset are divided into three subsets, the training set, the validation set and the testing set. The training set consists of 3,148 images from AFW (337), LFPW training set (811) and Helen training set (2,000). The validation set consists of 554 images including LFPW testing set (224) and Helen testing set (300). Following the existing works [28][25], two testing sets are employed: the IBUG dataset which includes 135 images, and the 300W competition dataset which includes 300 indoor images and 300 outdoor images. Following the setting of Zhu et al. [29], the AFLW dataset is used for training and evaluating the multi-view facial landmark detection.

Methods for comparison. To evaluate the effectiveness of our proposed FEC-CNN, it is compared with a few state-of-the-art methods including MDM [25], CFSS [28], ERT [14], CFAN [27], RCPR [3], SDM [26] and Zhu et al. [29]. For fair comparison, we follow the same testing settings as them, and directly refer to their released results [28][25][29].

All methods are evaluated in terms of the normalized root mean squared error (NRMSE) between the predicted landmark coordinates and the ground truth. For fair comparison with the existing methods, the NRMSE is normalized by the distance between the outer eye corners on 300W competition dataset, by the distance between the eye centers on 300W dataset and by the face size on the AFLW dataset. The cumulative error distribution (CED) and mean error are used for reporting performance.

B. Implementation Details of FEC-CNN

Data augmentation. To include more variations for better generalization, the training data are augmented by adding variance to each training sample, including rotation, translation, horizontal flipping, up-sampling and down-sampling.

Network structure. The FEC-CNN in all experiments consists of 3 sub-CNNs with the same structure to refine the shape. Instead of a mean shape S_0, we employ a deep CNN to directly predict an initial shape S_0 by taking the whole face image as input, in order to achieve more robustness to large global variations such as large poses, exaggerated expressions and partial occlusions. Although the CNN for initial shape is fixed in our experiment, it can be included in the end-to-end flow. The structure of CNN for initial shape and the structure of each sub-CNN in our FEC-CNN follow [16][9][10] and are respectively shown in Table I and Table II. In the table, the convolution layer is represented by C(n,k,g,s) where n, k, g and s denote the kernel number, kernel size, group number and stride respectively. The max pooling layer is represented by P(k,s) where k denotes the kernel size and s denotes the stride. The fully connected layer is represented by FC(n) where n denotes the hidden unit number. Furthermore, the size of the input image for

TABLE I

The structure of CNN for Initial Shape

Layer 1	Layer 2	Layer 3	Layer 4

C(24,11,1,4)
PReLU
LRN
P(3,2)
	C(64,5,2,1)
	PReLU
	LRN
	P(3,2)
		C(196,3,1,1)
		PReLU
			C(196,3,2,1)
			PReLU

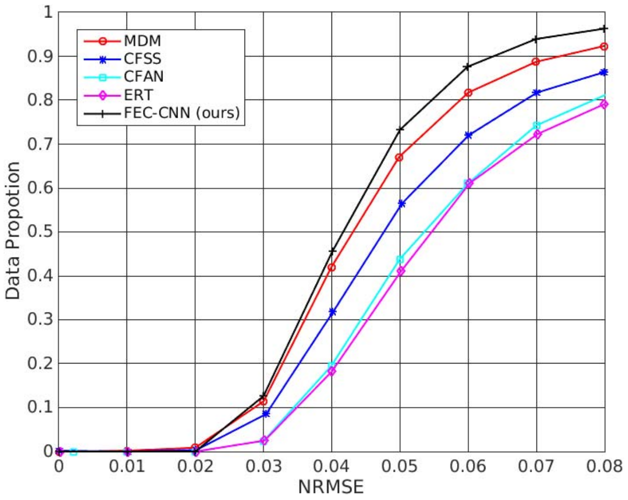
Layer 5	Layer 6	Layer 7	Layer 8
C(96,3,2,1) PReLU	FC(1024) PReLU Dropout	FC(1024) PReLU Dropout	FC(136)

TABLE II

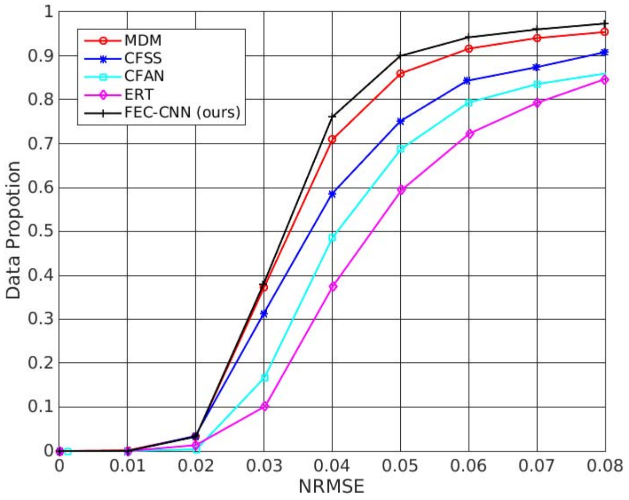
The Structure of each sub-CNN in the proposed FEC-CNN

Layer 1	Layer 2	Layer 3	Layer 4

C(16,6,1,2)
BN
ReLU
P(2,2)
	C(16,3,1,1)
	BN
	ReLU
	P(2,2)
		FC(128)
		BN
		ReLU
			FC(136)



(a) 68 points



(b) 51 points

Fig. 4. Comparison on 300W competition

initial CNN is 256*256, and the size of the shape-indexed patch in each sub-CNN is 31*31.

C. Evaluations on 300W competition

Firstly, we evaluate the performance of FEC-CNN, MDM [25], CFSS [28], CFAN [27] and ERT [14] on 300W competition dataset. Following the standard protocol, the performance of 68 points and 51 points is evaluated in terms of cumulative error distribution (CED). As seen from the results in Fig. 4, both our FEC-CNN and MDM [25] perform better than CFSS [28], CFAN [27] and ERT [14] benefited from the end-to-end learning structure. Moreover, our FEC-CNN outperforms the MDM which is attributed to the full consideration of the relationship between adjacent stages rather than only the relationship of the hidden layers of adjacent stages. As seen from the results of 68 points in Fig. 4(a), when NRMSE is 0.08, the data proportion of FEC-CNN is 96%, that is, few serious prediction error happens, which means FEC-CNN is robust for large global variations. Similar observations can be found in Fig. 4(b) demonstrating the superiority of the full consideration of the relationship of adjacent stages in our proposed FEC-CNN. Moreover, FEC-CNN preforms in 10 fps including the time consumption of the deep CNN for initial shape, and it can perform in real-time with a mean shape initialization.

D. Evaluations on 300W Dataset

Furthermore, the methods of CFSS [28], CFAN [27], RCPR [3], SDM [26] and our FEC-CNN are evaluated on 300W challenging subset (IBUG), which consists of 135 wild images with large poses, exaggerated expressions and partial occlusions. The performance of 68 landmark detection is shown in Fig. 5. As seen, the similar observations can be obtained that the proposed FEC-CNN achieves the best performance, demonstrating the effectiveness of FEC-CNN. Following the settings of [28], we also evaluate the mean error of FEC-CNN on 300W common subset and fullset, which is 0.042 and 0.049 respectively. Some exemplar results of landmark detection are shown in Fig. 6, from which we can see that the proposed FEC-CNN detects the landmarks accurately and is robust to large variations of pose, expression, lighting, occlusion and etc.

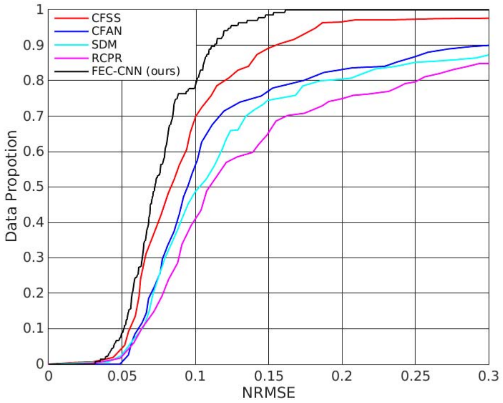


Fig. 5. Comparison on IBUG

E. Evaluations on AFLW Dataset

Moreover, we evaluate RCPR[3], Zhu et al. [29] and our FEC-CNN on a more challenging multi-view facial landmark dataset AFLW following [29]. The performance is reported in Table III. Our FEC-CNN outperforms the other methods which demonstrates its robustness and effectiveness for large pose and shape variation.

TABLE III
THE MEAN ERROR ON AFLW TESTSET

RCPR [3]	Zhu et al. [29]	FEC-CNN (ours)
0.037	0.027	0.017

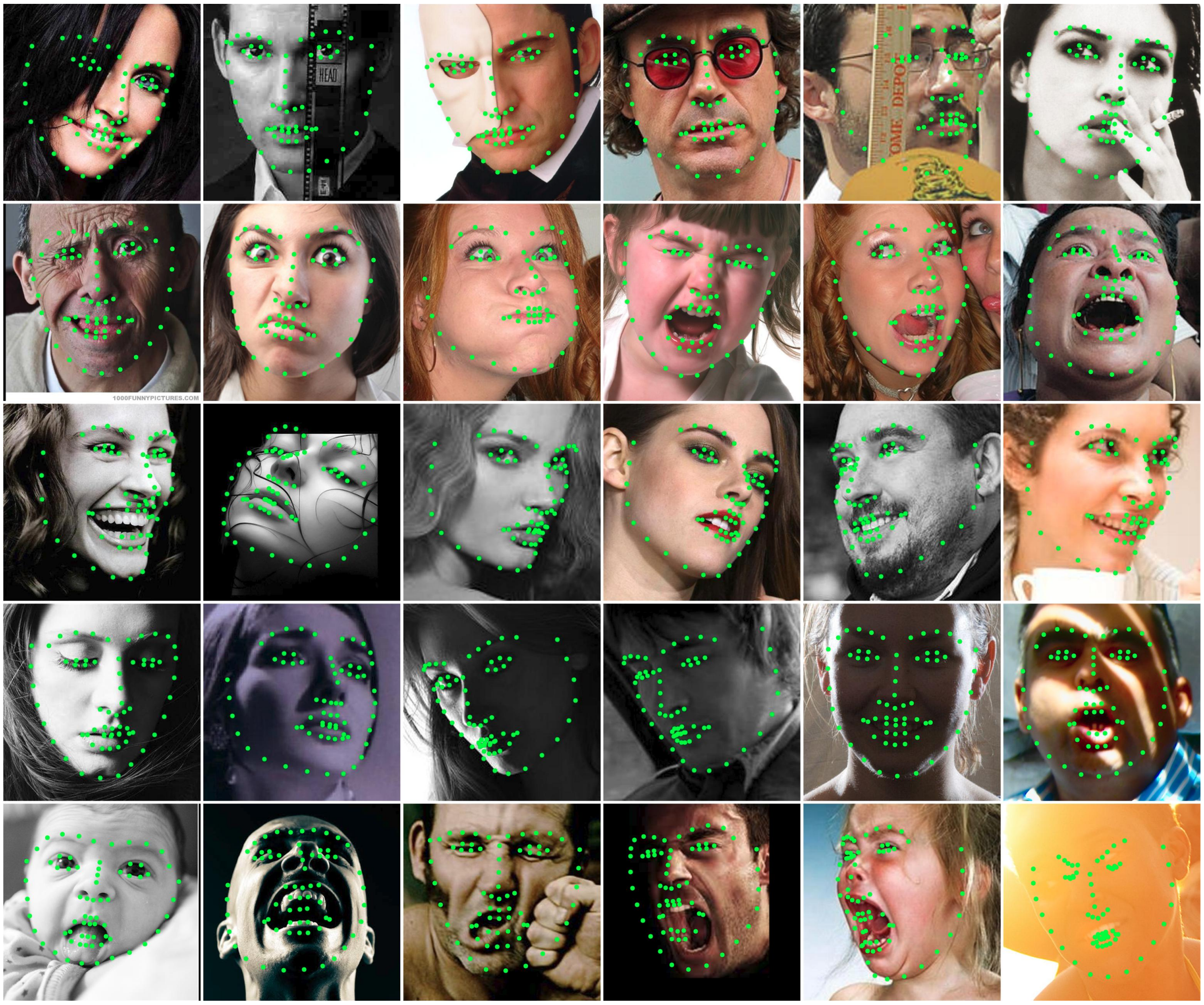


Fig. 6. Exemplar results of FEC-CNN prediction on IBUG and 300W competition. The first four rows contain samples with partial occlusion, large expression, large head pose and extreme illumination respectively. The last row shows some failure cases.

TABLE IV
THE MEAN ERROR ON IBUG

Mean shape	Initialization CNN
0.107	0.079

F. Evaluations on Initial Shape

To evaluate the influence of the initial shape on the performance, we use the mean shape and a initialization CNN for the initial shape respectively, and the mean errors on IBUG are shown in Table IV. As seen from the results, the initialization CNN significantly improves the performance of our FEC-CNN framework, which demonstrates the necessity of a robust initialization for better performance.

IV. Conclusions and Future Works

We propose a Fully End-to-End Cascaded CNN method for facial landmark detection problem. Our FEC-CNN fully considers the relationships between adjacent stages and is optimized end-to-end. Furthermore, FEC-CNN outperforms the state-of-the-art methods on three challenging datasets 300W competition, IBUG and AFLW. In future, we will try other network structures to further improve the prediction performance.

V. ACKNOWLEDGMENTS

This work was partially supported by 973 Program under contract No. 2015CB351802, Natural Science Foundation of China under contracts Nos. 61390511, 61650202, 61402443, 61272321, and the Strategic Priority Research Program of the CAS (Grant XDB02070004).

References

[1] M. S. Bartlett, G. C. Littlewort, M. G. Frank, C. Lainscsek, I. R. Fasel, and J. R. Movellan. Automatic recognition of facial actions in spontaneous expressions. *Journal of Multimedia (JMM)*, 1(6):22–35, 2006.

[2] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(12):2930–2940, 2013.

[3] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *International Conference on Computer Vision (ICCV)*, pages 1513–1520, 2013.

[4] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision (IJCV)*, 107(2):177–190, 2014.

[5] T. F. Cootes, G. J. Edwards, C. J. Taylor, et al. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(6):681–685, 2001.

[6] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding (CVIU)*, 61(1):38–59, 1995.

[7] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1078–1085, 2010.

[8] L. Gu and T. Kanade. A generative shape regularization model for robust face alignment. In *European Conference on Computer Vision (ECCV)*, pages 413–426, 2008.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[10] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[11] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2017–2025, 2015.

[12] A. Jourabloo and X. Liu. Pose-invariant 3d face alignment. In *International Conference on Computer Vision (ICCV)*, pages 3694–3702, 2015.

[13] A. Jourabloo and X. Liu. Large-pose face alignment via cnn-based dense 3d model fitting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1867–1874, 2014.

[15] M. Köstinger, 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 *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2144–2151, 2011.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[17] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *European Conference on Computer Vision (ECCV)*, pages 679–692, 2012.

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

[19] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision (IJCV)*, 60(2):135–164, 2004.

[20] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1685–1692, 2014.

[21] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: Database and results. *Image and Vision Computing (IVC)*, 47:3–18, 2016.

[22] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 397–403, 2013.

[23] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 896–903, 2013.

[24] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3476–3483, 2013.

[25] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 532–539, 2013.

[27] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *European Conference on Computer Vision (ECCV)*, pages 1–16, 2014.

[28] S. Zhu, C. Li, C. Change Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4998–5006, 2015.

[29] S. Zhu, C. Li, C.-C. Loy, and X. Tang. Unconstrained face alignment via cascaded compositional learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3409–3417, 2016.

[30] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3d solution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[31] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2879–2886, 2012.