Computer Standards & Interfaces 42 (2015) 105–112

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Adaptive Cascade Deep Convolutional Neural Networks for face alignment☆

Yuan Dong *, Yue Wu *

Beijing University of Posts and Telecommunications, 100876, PR China

ARTICLE INFO

Article history:
Received 27 April 2015
Received in revised form 8 June 2015
Accepted 8 June 2015
Available online 16 June 2015

Keywords:
Face alignment
Adaptive cascade
Deep convolutional networks
Gaussian distribution

ABSTRACT

Deep convolutional network cascade has been successfully applied for face alignment. The configuration of each network, including the selecting strategy of local patches for training and the input range of local patches, is crucial for achieving desired performance. In this paper, we propose an adaptive cascade framework, termed Adaptive Cascade Deep Convolutional Neural Networks (ACDCNN) which adjusts the cascade structure adaptively. Gaussian distribution is utilized to bridge the successive networks. Extensive experiments demonstrate that our proposed ACDCNN achieves the state-of-the-art in accuracy, but with reduced model complexity and increased robustness.

1. Introduction

Face alignment or facial landmark localization plays a critical role in many visual applications such as face recognition, face tracking, facial expression recognition and 3D face modeling. Therefore, it has been extensively studied in recent years. However, robust facial landmark detection remains a challenging problem when face images are taken under the situation with extreme occlusion, lighting, expressions and pose. To address this issue, research explores the modeling of shape variation and appearance variation for improved performance. In general, this type of research can be categorized into three groups: constrained local model based methods [2–4], active appearance model based methods [5,6] and regression based methods [1,7–10].

Constrained local models build classifiers called component detectors to search for each facial feature point independently. These component detectors calculate response maps to present the appearance variance around facial feature points. Due to the problems of ambiguity and corruption in local features, facial points detected by the local experts may be far away from the ground truth positions. Then shape constraints are applied to adjust the initial positions for improved results [2,4]. However, the global contextual information is difficult to be embedded into these methods.

Instead of modeling the appearance with each facial point, active appearance models such as Active Appearance Model (AAM) [5] use a holistic perspective to model the appearance variance. An AAM model is composed of a linear shape model and a linear texture model. The Principal Component Analysis (PCA) is applied to bridge the relationship between the two models. Nevertheless, simple linear models can hardly present the nonlinear variations of facial appearance in the case of faces taken in complex environment (e.g., extreme lighting).

Regression based methods, on the other hand, directly learn a regression function from image appearance (features) to the target output (shapes) [11]. Cascade architecture is usually employed and explored in regression based models. In each stage of the cascade architecture, shape-index features [12] are extracted to predict the shape increment with linear regression [7], tree-based regression [8] where the mean shape is used as the initializations of the shapes. Coarse-to-Fine Auto-Encoder Networks (CFAN) [9] utilizes a Stacked Auto-encoder Network [13] to predict the face shape quickly by taking a whole face as input. DCNN [1] employs a deep CNN model to extract high-level features to make accurate predictions as the initialization. After the initialization, the DCNN designs two-level convolutional networks to refine each landmark separately by taking local regions as input. To train these networks, several factors are critical for achieving good performance. For example, Sun et al. [1] conduct extensive experiments to investigate different network structures which are the basic regression units. The input range of local regions and the selecting strategy of local patches for training are other main factors having great impacts on the accuracy and reliability. But these factors are set by intuition or empirically in traditional methods. Besides, the relationship between any two successive networks is less developed.

In this paper, we propose an Adaptive Cascade Deep Convolutional Neural Networks (ACDCNN) for facial point detection. After initializing the shape by a CNN model like DCNN, each landmark is refined by a series of networks. These networks take the output of previous networks as input and locate a new position of the landmark. Different from

☆ The work is sponsored by the Chinese NSFC project 61372169.

* Corresponding authors.
E-mail addresses: yuandong@bupt.edu.cn (Y. Dong), wuyuebupt@gmail.com (Y. Wu).

http://dx.doi.org/10.1016/j.csi.2015.06.004
0920-5489/© 2015 Elsevier B.V. All rights reserved.

existing methods [1,9] which apply the same configuration of regression for each landmark or each facial component in a stage, we set the configurations according to different results of each landmark. In addition, a Gaussian distribution is used to model the output error of the previous network. The input range of the local region is related to the expectation and the standard deviation of this Gaussian distribution. After the input range is determined, patches centered at positions shifted from the ground truth position are taken for training. Instead of taking these patches randomly, they are fetched under that Gaussian distribution. Thus the most relevant image patches are selected for training the successive network. These better training samples lead to better performance. The comparison experiments show that the proposed ACDCNN outperforms or is comparable to the state-of-the-art methods on both robustness and accuracy.

The rest of the paper is organized as follows. Section 2 introduces related work followed by our proposed ACDCNN introduced in Section 3. The Implementation details are described in Section 4. Section 5 reports our experimental results followed by conclusion in Section 6.

2. Related work

Many approaches to face alignment have been reported in the past decades among which regression based methods show highly efficient and accurate performance thus have received increasing attentions. Valstar et al. [14] develop support vector regression to model the nonlinear transform from the input local features (Haar-like features) to target point locations. Dantone et al. [15] extend the regression forests [16] to conditional regression forests. Head poses are utilized in the framework as the prior probability. Cao et al. [17] use cascaded random ferns as regressors and take the shape-indexed features [12] extracted from the whole image as input. Ren et al. [8] propose the use of discriminative binary features with locality principle to further improve the accuracy and speed. Xiong et al. [7] develop the supervised descent method (SDM) with SIFT, and implement the regression procedure in a gradient descent view. Recently, an emerging field is deep models like convolutional networks widely used in computer vision applications such as image classification [18,19], object detection [20], scene recognition [21], face alignment [1,9,10] and face verification [22]. Sun et al. [1] propose the DCNN for point detection in a coarse-to-fine manner. Zhang et al. [9] develop Coarse-to-Fine Auto-Encoder Networks (CFAN), which utilizes several stacked auto-encoder networks to deal with the nonlinearity in inferring face shapes from face images. Zhang et al. [10] investigate the possibility of jointly optimizing facial landmark detection with a set of related task and propose a Tasks-Constrained Deep Convolutional Neural Network (TCDCNN).

We want to note that SDM [7] employs Normal distribution to generate training samples. However it only samples for initialization during training and the parameters of the Normal distribution capture the variance of a face detector. We argue while SDM may increase the robustness, the random sample in the whole process instead of only the initialization may further improve the robustness. Secondly, SDM uses the mean shape as the initialization to extract shape-indexed features, i.e. SIFT. The initialization is rough and may be far from the ground truth. In addition, SIFT is a hand-crafted feature which may be limited to complex shape with high nonlinearity. In DCNN [1], the configuration of input local patches and training parameters applied to each landmark is the same at the same level, which may ignore the different local appearance of each landmark. Secondly, DCNN selects the local training patches randomly by setting a maximum shift in both horizontal and vertical directions empirically, which lacks of intelligence guidelines. TCDCNN [10] jointly optimizes facial landmark detection with a set of related tasks requiring large number of labels of the data in training. CFAN [9] utilizes deep Auto-Encoder networks for the regression to model the complex nonlinearity between the SIFT features and the increment of the current shape. Deep Regression [23] makes use of a multistage structure based on linear regression and use back-propagation algorithm to jointly optimize the parameters. Both CFAN and Deep Regression use hand-crafted features which may suffer the same as that of SDM.

In this research, we propose a novel Adaptive Cascade Deep Convolutional Neural Networks (ACDCNN), which is based on DCNN, augmented with sampling strategies in every cascade of two successive networks. The advantages are two-fold. First, ACDCNN utilizes the whole face as input to predict the initial face shape, which is employed in DCNN. The global high-level feature extracted with a new deep network structure is capable to handle faces taken under complex environment. Secondly, ACDCNN models the error of the previous output with a Normal distribution and selects these patches under this distribution in any two consecutive networks (instead of only in the initiation). This sampling strategy can refine the network structure and its parameters adaptively thus improved robustness is expected.

3. Adaptive Cascade Deep Convolutional Neural Networks

In this section, we present a novel method termed ACDCNN for face alignment. The details of each component of ACDCNN, including the initialization with a Deep Convolutional Neural Network and the Local Adaptive Cascade Networks (LACN) are explained.

3.1. Method overview

As shown in Fig. 1, the proposed ACDCNN attempts to design a unified cascade pipeline for each facial point, with the regression in each stage modeled as a deep convolutional network. There are five facial points to detect: *left eye center* (LE), *right eye center* (RE), *nose tip* (N), *left mouth corner* (LM), and *right mouth corner* (RM). At the first level, a deep convolutional network is employed to predict all five facial landmarks simultaneously. The whole face is taken as input and high-level features learned from raw RGB pixels are utilized to make predictions. After getting an estimation of face shape from the first level, each facial landmark is refined by a set of networks. These networks take local patches centered at the predicted positions of the facial point from previous levels as input. Each set of networks for facial points is independent because local appearance of each landmark is different. And the error of each point in previous networks is distinguished from each other. To characterize the variations, each output error is modeled by Gaussian distribution. The following networks take the expectation and the standard deviation of this Gaussian distribution into consideration to estimate a better configuration for training. The iteration will stop once the error is no longer reduced.

3.2. Initialization with DCNN

The first network directly estimates the face shape by taking the whole face as input. Given a face image $x \in \mathrm{R}^{m \times 1}$ of m pixels, $\mathrm{d(x)} \in \mathrm{R}^{2p \times 1}$ denotes p landmarks ($p = 5$) in the face image. The task of facial landmark detection is to learn a mapping function F from the image to the face shape as follows:

$$F : d(x) \leftarrow x.$$

Due to the complexity and nonlinearity of the mapping function, a deep convolutional network is developed. Raw RGB pixels are taken as input by the network and coordinates of the five points (one for each landmark) are the outputs. The architecture of the network contains six learned layers — three convolutional and three fully-connected. Each convolutional layer applies square convolution kernels (or filters) to the multichannel input feature maps and output the responses. Specifically, let the input layer be $I(h, w, l)$, where h, w and l are the height, width and depth of the input. Convolutional layer is denoted by $F(s, k, n)$, where s is the side length of the convolution kernel, k is the depth of the convolution kernel, n is the number of kernels in the

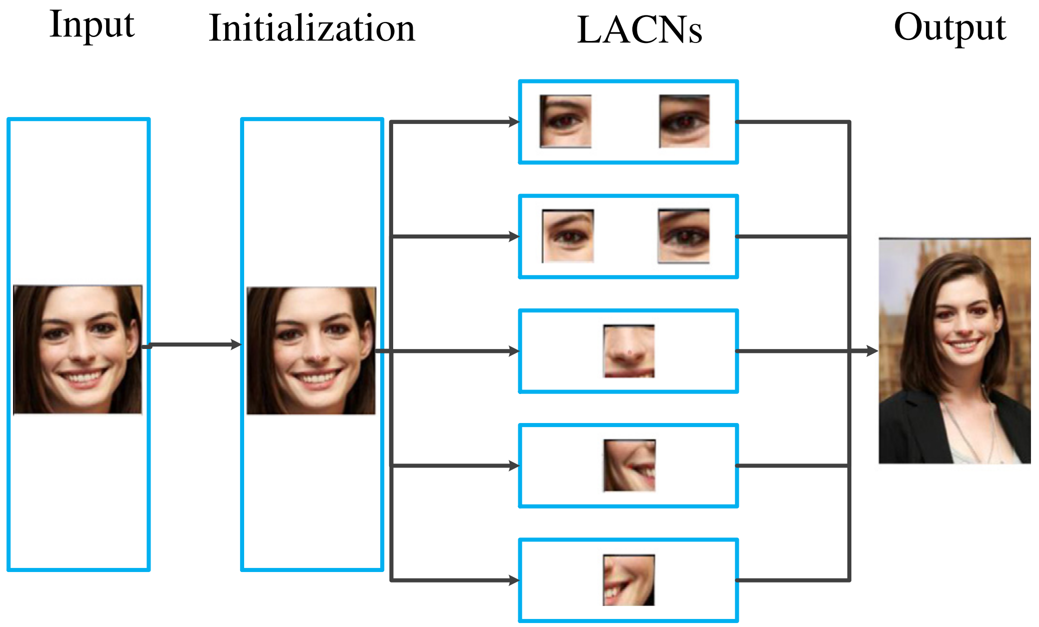


Fig. 1. System overview. The input is the face region which is slightly expended from the original region detected by a face detector. The first network predicts five landmarks simultaneously for initialization. Local Adaptive Cascade Networks (LACN) is a series of networks which are used to locate a facial landmark in local regions. Five LACNs are used to refine predictions of each landmark independently. The final output consists of positions of all facial points.

convolutional layer. The response of a kernel in the position (i, j) of the output layer is computed as:

$$y_{i,j} = \max\left(\sum_{x=0}^{s-1}\sum_{y=0}^{s-1}\sum_{z=0}^{k-1} I_{i+x,j+y,z} \cdot W_{x,y,z} + B, 0\right).$$

Where x, y, z are the coordinates of a position in the convolution kernel, and W and B are weight and bias. Instead of the standard hyper-tangent function $f(x) = \tanh(x)$, Rectified Linear Units (ReLUs) [18] $f(x) = max(0, x)$ is applied to the filter responses to accelerate the training procedure.

After convolution, max-pooling with overlapping regions is used:

$$y_{i,j} = \max_{0 \le x,y < q} \left\{ I_{i \cdot p+x, j \cdot p+y} \right\}$$

where q is the side length of square pooling regions and p is the stride size between successive pooling squares. $p < q$ is set to obtain overlapping pooling.

The final prediction is produced by three fully connected layers. Euclidean loss is used to regress real-valued labels as following:

$$F = \underset{F}{\operatorname{argmin}} \, \|F(x) - label(x)\|_2^2.$$

After the optimization, the prediction of the facial landmarks is achieved as S_0, which is the initialization of all landmarks.

More details of the architecture are shown in Fig. 2.

3.3. Local Adaptive Cascade Networks

Although the deep convolutional network is able to model the nonlinear mapping from raw pixels of a face image to a face shape (via landmarks), there is still undesirable gap between the estimated shape and the ground truth location due to the highly complicated variations in pose, expression, illumination, etc. As a result, the face shape needs to be refined by successive deep convolutional networks. Since the global information is taken account in the initialization step, each facial point is optimized locally and independently in the subsequent networks. Local Adaptive Cascade Networks is designed to achieve this purpose.

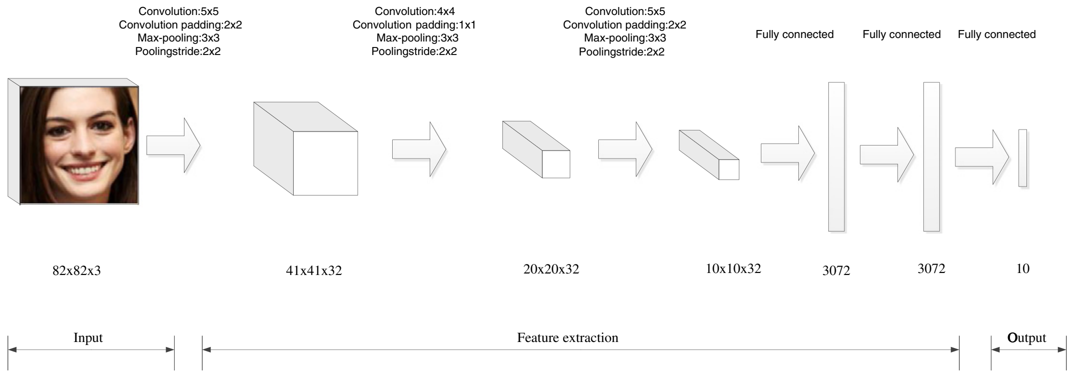


Fig. 2. Typical structure of networks in our system. The network consists of convolutional layers, max-pooling layers and fully-connected layers. Max-pooling is performed after convolutional layers with overlapping. The architectures of other networks are similar.

Specifically, to capture the subtle appearance around a landmark, local patches centered at the predicted position from the previous level are taken as input. The position of a landmark is denoted by $p_{old} \in \mathrm{R}^{2\times 1}$. Given a local patch $x(p_{old}) \in \mathrm{R}^{n\times 1}$ of n pixels, $p_{new} \in \mathrm{R}^{2\times 1}$ is the new position predicted in the local region. Similarly as the first stage, a deep convolutional network is employed to deal with the complex nonlinearity of the mapping function with each network containing three convolutional layers followed by max pooling, and three fully connected layers.

Selecting the local patches is an important factor to train the network. The training procedure aims at correcting the position p_{new} of the landmark in a local region centered at the previous position p_{old}. Thus, the output of the previous network is highly related with the patches selecting of the current network. Let p_{gt} denotes the ground truth positon of a landmark. A Gaussian distribution $f(x, u, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-u)^2}{2\sigma^2}}$ is used to model the error:

$$\|p - p_{gt}\|_2.$$

During training, positions shifted from the ground truth position are selected to be the center of training patches under this Gaussian distribution. The offset $(\Delta x, \Delta y)$ is calculated by:

$$\Delta x = \begin{cases} -\Delta x_{max}, & r\cdot\cos\theta \le -\Delta x_{max} \\ r\cdot\cos\theta, & -\Delta x_{max} \le r\cdot\cos\theta \le \Delta x_{max} \\ \Delta x_{max}, & r\cdot\cos\theta \ge \Delta x_{max} \end{cases}$$
$$\Delta y = \begin{cases} -\Delta y_{max}, & r\cdot\sin\theta \le -\Delta y_{max} \\ r\cdot\sin\theta, & -\Delta y_{max} \le r\cdot\sin\theta \le \Delta y_{max} \\ \Delta y_{max}\ , & r\cdot\sin\theta \ge \Delta y_{max} \end{cases}$$

where r is a radius sampled from the distribution $f(x, u, \sigma)$, θ is a random angle which determines the direction of the center, and Δx_{max}, Δy_{max} are the max shift in horizontal direction and vertical direction, respectively. Δx_{max} and Δy_{max} are set by:

$$\Delta x_{max} = \mu + 2\sigma$$
$$\Delta y_{max} = \mu + 2\sigma.$$

Then the center of the selected patch is denoted by $p_{gt} + (\Delta x, \Delta y)$. This is the Adaptive Cascade, which connects two successive networks. The parameter “2” is set empirically.

Before a local patch is fed into the convolutional network, it needs to be cropped from the original face image and be resized to a fixed size. Local patches which have large input ranges contain more context information. By contrast, small input ranges constrain the local patches in small regions which have more details and lead to more accurate predictions. Thus, the input range varies with the current accuracy of each landmark. Let Δw donates the width of the square patches, and it is set as following:

$$\Delta w = 10 \times \Delta x_{max}.$$

Note that the same configuration to select the patches of each landmark should be maintained during testing.

4. Implementation

4.1. Structures

Networks at different levels follow a similar architecture with varied numbers of the inputs. Table 1 summarizes the input sizes for different facial components. All networks are trained on raw RGB values of the pixels. The networks used in the first level, the left eye and right eye have higher resolution since it is observed the whole faces and the regions of eyes contain richer context information than the regions of the nose and mouth.

Table 1
Input resolution, filter size and number of channels of the networks. INIT is used to locate five landmarks for initialization. EYE represents the networks used for the left eye center and the right eye center in the system. OTHER is used for the nose, the left mouth corner and the right mouth corner.

	INIT	EYE	OTHER
Input	82 × 82 × 3	82 × 82 × 3	40 × 40 × 3
Conv. 1	5 × 5 × 32	5 × 5 × 32	5 × 5 × 32
Conv. 2	4 × 4 × 32	4 × 4 × 32	4 × 4 × 32
Conv. 3	5 × 5 × 32	5 × 5 × 32	5 × 5 × 32
Hidden 1	3072	3072	3072
Hidden 2	3072	3072	3072
Output	10	2	2

4.2. Training

All networks are trained by stochastic gradient descent with hand-tuned learning rate. The dropout technique [24] is applied to the first two fully-connected layers for reducing complex co-adaptations of neurons, which the output of each hidden neuron is set to zero with probability 0.2. And image patches are pre-processed by subtracting the mean value over the training set from each pixel.

4.3. Data augmentation

Training patches are augmented by slight transformation (translation and rotation) before feeding into networks in order to reduce overfitting. At the first level, training patches are taken according to face bounding boxes. We augment these patches by a random shift from the center of a face bounding box. The maximum shift in both horizontal and vertical directions is 0.1, where the distance is normalized with the face bounding box. For the other networks, translation is not applied since the center of them is selected under certain Gaussian distribution. Besides, each patch is rotated to a random angle in the range (−5°, 5°).

5. Experiments

In this section, we firstly illustrate the datasets for the evaluations. Every cascade network for each facial landmark in our method is investigated. Next, the comparison with DCNN is conducted on the same training and validation set. Finally, we compare the proposed ACDCNN with the state-of-the-art methods and commercial software.

5.1. Datasets

The first dataset with 13,466 face images [1] is utilized in our experiments for training and validation. The dataset contains 5590 LFW images and 7876 images downloaded from the web. Following [1], 10,000 images are selected for training and the remaining 3466 images are used for validation.

The second dataset is BioID [25] which contains 1521 gray level images with a resolution of 384 × 286. These face images are from 23 different persons. All the faces have limited variations on pose, expression and illumination.

To test the robustness of the system, the third dataset is LFPW [2] which includes 1432 face images, 1132 images for training and 300 images for testing. These images are downloaded from web and the faces show large variations on pose, illumination and expression. Partly occlusion also is observed in this dataset.

For all three datasets, each face image contains a bounding box and five landmarks. And all images for training and validation are also labeled under the same condition.

Following [1], performance is measured using mean error and failure rate of each facial landmark. The mean error is measured by the distance between the predicted landmark position and the ground truth position

Table 2
The μ of all results on the validation set.

	LE	RE	N	LM	RM
μ_0	0.0132	0.0129	0.0165	0.0164	0.0162
μ_1	0.0082	0.0081	0.0094	0.0111	0.0112
μ_2	0.0077	0.0075	–	–	–

Table 3
The σ of all results on the validation set.

	LE	RE	N	LM	RM
σ_0	0.0097	0.0094	0.0105	0.0112	0.0112
σ_1	0.0081	0.0076	0.0069	0.0099	0.0099
σ_2	0.0081	0.0073	–	–	–

Table 4
The Δx_{max}, Δy_{max} for the seven local convolutional networks.

	LE	RE	N	LM	RM
$\Delta x_{max,1}$, $\Delta y_{max,1}$	0.0326	0.0316	0.0375	0.0389	0.0386
$\Delta x_{max,2}$, $\Delta y_{max,2}$	0.0244	0.0234	–	–	–

Table 5
The Δw for the seven local convolutional networks.

	LE	RE	N	LM	RM
Δw_1	0.3260	0.3164	0.3748	0.3888	0.3856
Δw_2	0.2438	0.2342	–	–	–

normalized by inter-ocular distance or face width, which can be formulated as

$$\frac{\sqrt{\left(x-x_{gt}\right)^2+\left(y-y_{gt}\right)^2}}{w}.$$

Where (x, y) and (x_{gt}, y_{gt}) are the detected position and the ground truth, and w is the inter-ocular distance or face width. Mean error larger than a threshold is counted as a failure. For fair comparison, we use the width of the face bounding box to normalize the mean error in Sections 5.2 and 5.3 and will switch to inter-ocular distance in Section 5.4. And the threshold to report failure is set as 5% in Sections 5.2 and 5.3 and 10% in Section 5.4.

5.2. Investigation of Adaptive Cascade

As the proposed ACDCNN method consists of a global convolutional network and several independent LACN, we investigate how each LACN reduces the mean error of each landmark. The experiments are conducted on the training set and the validation set to the same as [1].

The statistical information of the results of the different stages on validation set is shown in Tables 2–3. The corresponding training configuration is illustrated in Tables 4–5. Taking the first local network of LE (left eye center, one of the five landmarks) as an example, the max shifts and the input ranges of it are calculated:

$$\Delta x_{max} = \Delta y_{max} = 0.0326$$

$$\text{Input range} = 0.1630.$$

A deep convolutional network is trained to predict the LE position by taking local patches around the LE as input. The distance r between the center of the local patch and the ground truth of LE is sampled under the distribution:

$$f(r) = \frac{1}{0.0097 \times \sqrt{2\pi}} e^{-\frac{(r-0.0132)^2}{2\times 0.0097^2}}.$$

During the testing phase, the network trained for LE outputs a new position of LE in the local patch centered at the previous position. Similarly, a new deep convolutional network is employed to further reduce the mean error of LE. The iteration will stop when the mean error is no longer decreased. Networks for other landmarks are trained in a similar way. The whole system consists of eight convolutional networks.

The evaluation results on validation set are shown in Fig. 3. As seen, although adaptive cascade networks only iterate for one stage and utilize only one network, the performance is improved significantly. The mean error is reduced by 37.9% and 6.1% in the two iterations for LE. The mean error of RE is reduced by 37.2% and 7.4% after two refinements. The improvement of N, LM and RM is 43.3%, 32.3% and 30.9%, respectively. The failure rate shows the similar tendency. But after the second local refinement of LE, the failure rate is slightly worse than the previous result. This is because the refinement brings slight drift to the points which are well located by the previous networks. But the mean error is still reduced. We conclude this network maintains the ability to refine the current locations of the landmarks.

5.3. Comparison with the DCNN [1]

Although both ACDCNN and the DCNN are built on CNN and follow the cascade framework, we show that the proposed system can achieve better performance and dramatically reduce the system complexity. Using the same 3466 images from [1], we compare ACDCNN against the DCNN [1].

We take the initialization results and the final results of these two methods for comparison. Note that we use the same 10,000 training faces as in the DCNN. Thus the difference is that we exploit a different network structure and an adaptive manner for training. As seen in Fig. 4, the mean error of the initial results for five facial landmarks of both methods is nearly the same except the nose. ACDCNN has a worse initialization of nose. However, our ACDCNN performs better in all five facial landmarks in the final results, both in mean error and in failure rate. In addition, it shall be noted that only eight convolutional networks are included in ACDCNN while there are twenty three networks in the DCNN. We conclude our system is more succinct than the DCNN. Some detection results on the validation set are shown in Fig. 8.

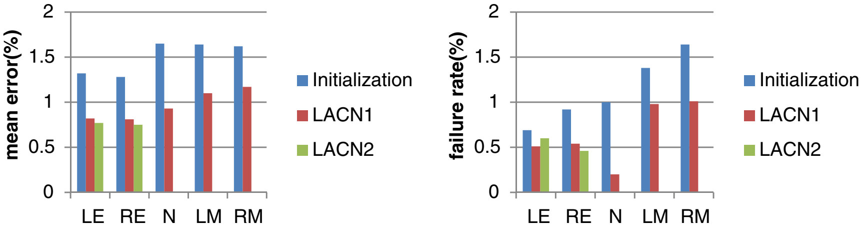


Fig. 3. Average detection errors and failure rates of all networks for each landmark.

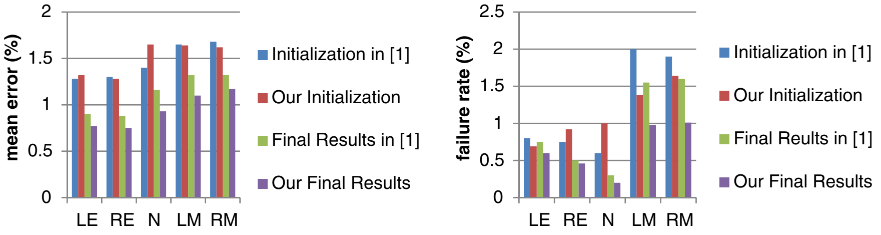
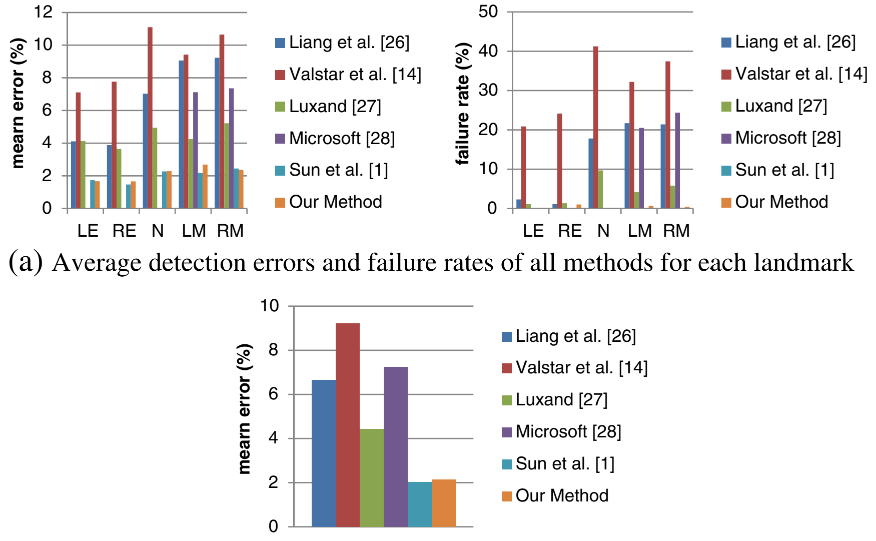


Fig. 4. Comparison between ACDCNN and DCNN.



(a) Average detection errors and failure rates of all methods for each landmark

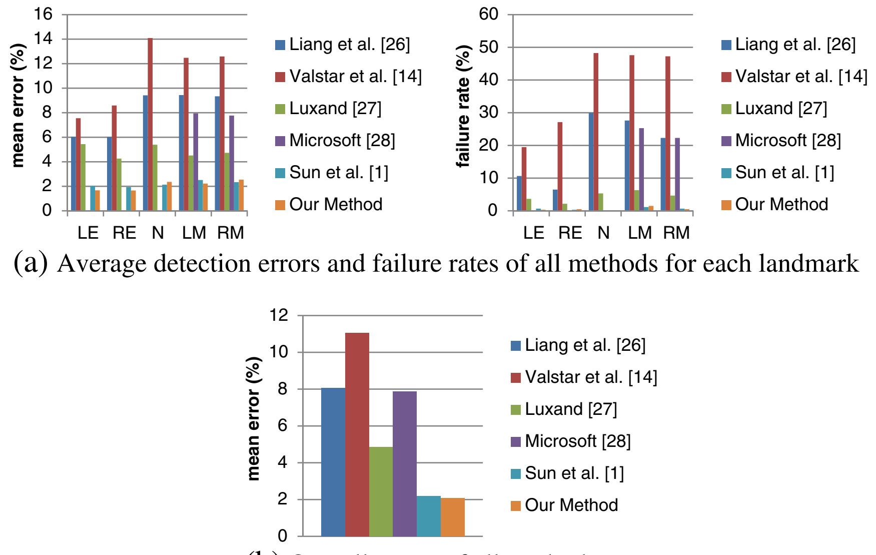
(b) Overall errors of all methods.

Fig. 5. Comparison on BioID.

5.4. Comparison with other state-of-the-art methods and commercial systems

In this section, we compare ACDCNN against: (1) Component based Discriminative Search [26], (2) Boosted Regression with Markov Networks [14], (3) Luxand Face SDK [27], (4) Microsoft Research Face SDK [28], (5) the algorithm using a consensus of exemplars (CoE) [2], and (6) Explicit Shape Regression (ESR) [17]. Following the same experimental setting as in [1], the evaluation is conducted using BioID and LFPW datasets. Since the Sun et al. [1] report the results of DCNN on these images, it is also included for comparison.

The comparison results of BioID are shown in Fig. 5. Due to the face detection failure, only 1341 images are utilized to report the results. Note that Microsoft Face SDK does not return the positions of eye centers and



(a) Average detection errors and failure rates of all methods for each landmark

(b) Overall errors of all methods

Fig. 6. Comparison on both LFPW training and test images.

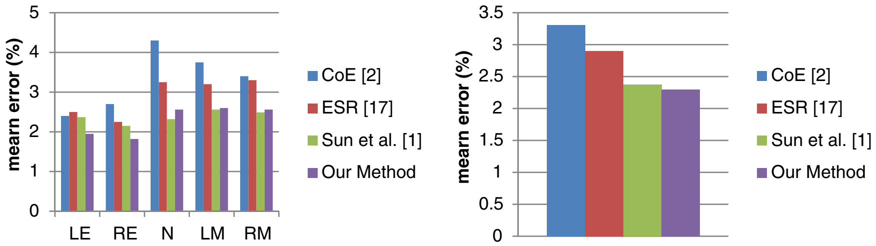


Fig. 7. Comparison on LFPW test images. The left subfigure shows the mean errors on different landmarks, while the overall errors are shown in the right subfigure.

nose. Our method outperforms the DCNN on LE and RM but underperforms on RE, N and LM. For the overall errors, our method has inferior performance to a small degree. We then make further investigation on the dataset. It is observed that all images of this dataset are gray-scale images. Since our model is trained on color images, we copy the gray channel of images three times before feeding them into the system during testing. As a result, we observe our method has comparable results to the DCNN and outperforms other methods. Results are shown in Fig. 8.

Similarly, for LFPW dataset, 601 images out of 1030 images are tested. Fig. 6 shows that our method outperforms all the other methods including the DCNN. Fig. 8 presents some detection examples using our model. We observe that the proposed method is able to handle images that contain great variation in pose, lighting and severe occlusion.

In addition, CoE, ESR and the DCNN report results on LFPW test images. The result of CoE is on all 300 test images while the result of ESR and the DCNN is on 249 images because of the disappearing of image URLs. Our model is also evaluated on these 249 images to be compared with the three state-of-the-art methods. The mean error of each landmark and the overall errors are shown in Fig. 7. As seen, our proposed method has the best performance.

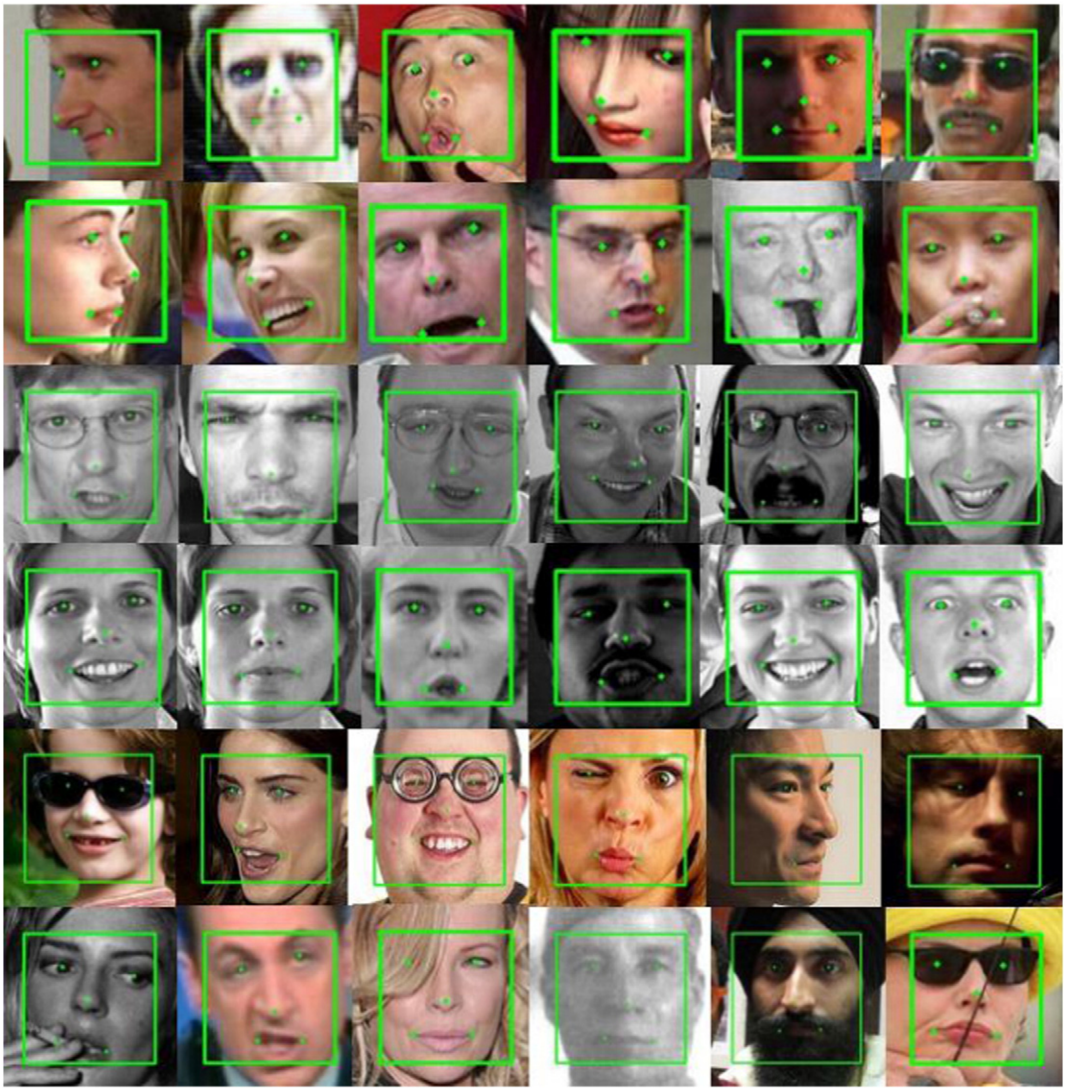


Fig. 8. Our results on the validation set, BioID and LFPW.

6. Conclusions and future work

In this paper, an adaptive cascade framework for face alignment is proposed. Gaussian distribution is used to bridge the current network input with the previous network output. Each landmark is refined independently based on its previous statistical information on which the adaptively sampling strategy to select training patches depends. The benefit of adaptive sampling lies in that the most relevant image patches are exploited for training deep convolutional neural networks. We show that the system can achieve comparable or better performance on three datasets. Moreover, our system with 8 networks is more succinct than the DCNN which consists of 23 networks. In future work, we plan to locate extensive facial landmarks with similar principle for further improved performance.

References

[1] Y. Sun, X. Wang, X. Tang, Deep convolutional network cascade for facial point detection, Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE June 2013, pp. 3476–3483.
[2] P.N. Belhumeur, D.W. Jacobs, D. Kriegman, N. Kumar, Localizing parts of faces using a consensus of exemplars, Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE June 2011, pp. 545–552.
[3] T.F. Cootes, M.C. Ionita, C. Lindner, P. Sauer, Robust and accurate shape model fitting using random forest regression voting, Computer Vision—ECCV 2012, Springer, Berlin Heidelberg 2012, pp. 278–291.
[4] X. Zhu, D. Ramanan, Face detection, pose estimation, and landmark localization in the wild, Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE June 2012, pp. 2879–2886.
[5] T.F. Cootes, G.J. Edwards, C.J. Taylor, Active appearance models, Computer Vision—ECCV'98, Springer, Berlin Heidelberg 1998, pp. 484–498.
[6] X. Gao, Y. Su, X. Li, D. Tao, A review of active appearance models, Syst. Man Cybern. C Appl. Rev. IEEE Trans. 40 (2) (2010) 145–158.
[7] X. Xiong, F. De la Torre, Supervised descent method and its applications to face alignment, Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE June 2013, pp. 532–539.
[8] S. Ren, X. Cao, Y. Wei, J. Sun, Face alignment at 3000 fps via regressing local binary features, Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE June 2014, pp. 1685–1692.
[9] J. Zhang, S. Shan, M. Kan, X. Chen, Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment, Computer Vision—ECCV 2014, Springer International Publishing 2014, pp. 1–16.
[10] Z. Zhang, P. Luo, C.C. Loy, X. Tang, Facial landmark detection by deep multi-task learning, Computer Vision—ECCV 2014, Springer International Publishing 2014, pp. 94–108.
[11] N. Wang, X. Gao, D. Tao, X. Li, Facial Feature Point Detection: a Comprehensive Survey. arXiv, preprint arXiv:1410.1037, 2014.
[12] P. Dollár, P. Welinder, P. Perona, Cascaded pose regression, Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE June 2010, pp. 1078–1085.
[13] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[14] M. Valstar, B. Martinez, X. Binefa, M. Pantic, Facial point detection using boosted regression and graph models, Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE June 2010, pp. 2729–2736.
[15] M. Dantone, J. Gall, G. Fanelli, L. Van Gool, Real-time facial feature detection using conditional regression forests, Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE June 2012, pp. 2578–2585.
[16] L. Breiman, Random forests[J], Mach. Learn. 45 (1) (2001) 5–32.
[17] J. Sun, F. Wen, Y. Wei, et al., Face alignment by Explicit Shape Regression[C]//2013 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012. 2887–2894.
[18] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet Classification With Deep Convolutional Neural Networks[C]//Advances in Neural Information Processing Systems, 2012. 1097–1105.
[19] C. Szegedy, W. Liu, Y. Jia, et al., Going Deeper With Convolutions[J]. arXiv, preprint arXiv:1409.4842, 2014.
[20] R. Girshick, J. Donahue, T. Darrell, et al., Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[C]//Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014. 580–587.
[21] B. Zhou, A. Lapedriza, J. Xiao, et al., Learning Deep Features for Scene Recognition Using Places Database[C]//Advances in Neural Information Processing Systems, 2014. 487–495.
[22] Y. Sun, X. Wang, X. Tang, Deep Learning Face Representation From Predicting 10,000 Classes[C]//Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, IEEE, 2014. 1891–1898.
[23] B. Shi, X. Bai, W. Liu, et al., Deep Regression for Face Alignment[J]. arXiv preprint arXiv:1409.5230, 2014.
[24] G.E. Hinton, N. Srivastava, A. Krizhevsky, et al., Improving Neural Networks by Preventing Co-adaptation of Feature Detectors[J]. arXiv, preprint arXiv:1207.0580, 2012.
[25] O. Jesorsky, K.J. Kirchberg, R.W. Frischholz, Robust Face Detection Using the Hausdorff Distance[C]//Audio-and Video-based Biometric Person Authentication, Springer, Berlin Heidelberg, 2001. 90–95.
[26] L. Liang, R. Xiao, F. Wen, et al., Face Alignment Via Component-based Discriminative Search[M]//Computer Vision–ECCV 2008, Springer, Berlin Heidelberg, 2008. 72–85.
[27] http://www.luxand.com/facesdk/.
[28] http://research.microsoft.com/en-us/projects/facesdk/.

Dong Yuan is associate professor at Beijing University of Posts and Telecommunications, China. He is also invited as "France Telecom — Orange Expert on Solution of Content Service" of France Telecom R&D Group. He received his PhD degree in Shanghai Jiao Tong University at 1999, worked as R&D scientist at Nokia Research Center China from 1999–2001, worked as post-doctoral research staff at Engineering Department Cambridge University UK from 2001–2003. His current research interests include semantic video indexing, video copy detection, and multimedia content search.

Wu Yue is a postgraduate student at Beijing University of Posts and Telecommunications, China. He received the B.S. degree in Electronic Information Engineering in Beijing University of Posts and Telecommunications at 2013. His current research interests are face tracking, face alignment, face recognition, object detection and deep learning.