Neurocomputing 273 (2018) 222–229

Contents lists available at ScienceDirect

Neurocomputing

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

Facial landmark localization by enhanced convolutional neural network

Weihong Deng[a,*], Yuke Fang[b], Zhenqi Xu[a], Jiani Hu[a]

[a] School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
[b] International School, Beijing University of Posts and Telecommunications, Beijing 100876, China

ARTICLE INFO

Article history:
Received 7 December 2016
Revised 10 July 2017
Accepted 20 July 2017
Available online 22 August 2017

Communicated by Dr. Tie-Yan Liu

Keywords:
Face recognition
Face alignment
Facial landmark localization
Convolutional neural network
Deep learning

ABSTRACT

Facial landmark localization is important to many facial recognition and analysis tasks, such as face attributes analysis, head pose estimation, 3D face modeling, and facial expression analysis. In this paper, we propose a new approach to localizing landmarks in facial image by deep convolutional neural network (DCNN). We make two enhancements on the CNN to adapt it to the feature localization task as follows. First, we replace the commonly used max pooling by depth-wise convolution to obtain better localization performance. Second, we define a response map for each facial points as a 2D probability map indicating the presence likelihood, and train our model with a KL divergence loss. To obtain robust localization results, our approach first takes the expectations of the response maps of enhanced CNN and then applies auto-encoder model to the global shape vector, which is effective to rectify the outlier points by the prior global landmark configurations. The proposed ECNN method achieves 5.32% mean error on the experiments on the 300-W dataset, which is comparable to the state-of-the-art performance on this standard benchmark, showing the effectiveness of our methods.

1. Introduction

Face alignment or facial landmarks detection is to automatically find salient facial points such as the center of the pupils, the tip of the nose, the corners of the mouth and so on. It has attracted much interests due to its importance for many facial analysis applications, e.g. feature based face recognition [1], head pose estimation [2], 3D face modeling [3] and facial expression analysis [4]. Our goal is to localize a large number of predefined landmarks in the images taken under a variety of acquisition conditions, including pose, lighting, expression, hairstle, subject age, ethnicity, and partial occlusion. Even though many methods have been developed for face alignment recently, this problem still remains an open issue due to large pose variations and complex expression conditions. Besides, occlusion caused by eyeglasses or hair makes it even harder. It is difficult for a model to predict a location if the edges or corners are occluded.

We work on localizing the facial landmarks through detecting the fine-scale fiducial points followed by global shape constraints. Fiducial point detectors typically include binary classifiers, such as support vector machines (SVMs) [5] and adaboost cascade [6], that are trained to respond to a specific fiducial (e.g., eye centers or mouth corners). These local detectors often convolute the image with a bank of filters (e.g., wavelets [5], Gaussian derivative filters [7], Gabor filters [14], [8], or Haar-like features [9]). These local detectors are scanned over a portion of the image and may return one or more candidate locations for the part or a score at each location. To reduce the false detections and boost the efficiency, feature detectors mostly search over a smaller region that includes the actual part location with minimal impact of missing fiducials [9]. However, even for well-trained classifiers, false detections often occur because different facial portions can have the appearance of the same fiducial under unconstrained conditions.

To address this problem, we propose to solve the landmark detection problem by mapping input facial image to the response map using deep convolutional neural network (DCNN) [10]. The original CNN architecture is designed for image classification and may not be suitable for mapping regression. Thus we make two enhancements on the CNN as follows. First, we replace the commonly used max pooling by depth-wise convolution to obtain better location performance. Second, we define a response map for each facial point, which is a 2D probability map indicating the presence likelihood of facial points at the corresponding locations, and train our model from scratch with a KL divergence loss. To obtain robust localization results, we first consider the expectations of the response maps of enhanced CNN and then apply auto-

* Corresponding author.
E-mail addresses: whdeng@bupt.edu.cn, cvpr_dwh@126.com (W. Deng).

http://dx.doi.org/10.1016/j.neucom.2017.07.052
0925-2312/

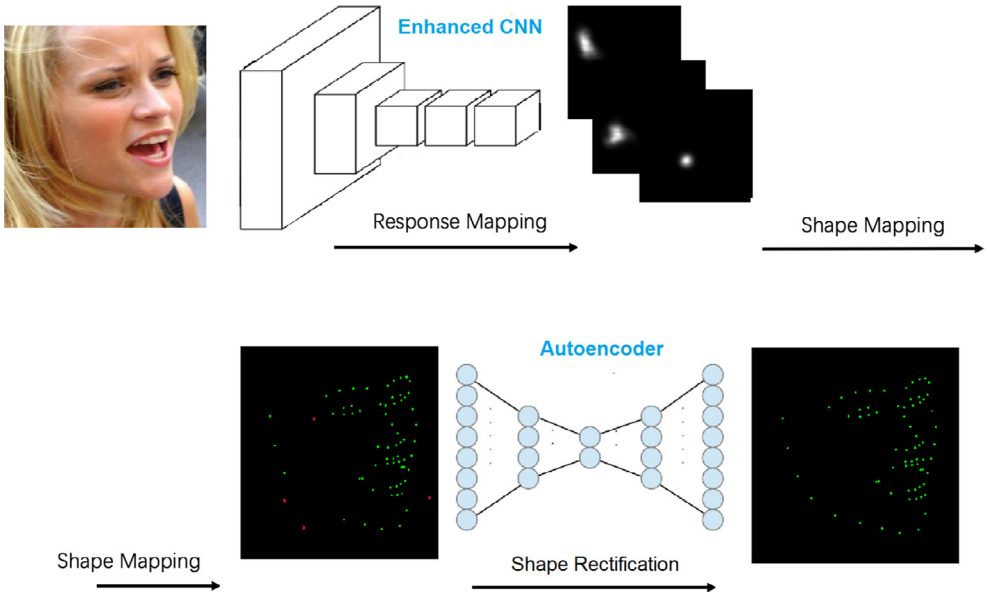


Fig. 1. There are three stages of our face alignment method: response mapping, shape mapping, and reshape rectification. In response mapping stage, we propose an enhanced convolutional neural network to learn the response maps for feature points. In shape mapping stage, we compute expectation location of each feature point according to the corresponding feature map. Finally, the shape rectification stage applies correct the outlier feature points by the auto-encoder that encodes the global shape constraint.

encoder model on it, which is effective to reduce the noise by the prior global information about the feature configurations. The overall architecture can be seen in Fig. 1. The proposed ECNN method is successfully tested on the experiments on the 300-W dataset [11], and achieves 5.32% mean error rate, which is comparable to the state-of-the-art performance on this standard benchmark, showing the effectiveness of our methods.

2. Related works

Recently, there are a number of unconstrained face alignment methods having been proposed, mainly including local response based method, cascaded regression based method and deep learning based method.

2.1. Local response based method

The local response based method usually contains two stages. In the first stage, a model is learned for each facial point to predict the probability whether a location is a key point. In the second stage, a global shape model is implemented to constrain the relation between points. Belhumeur et al. [12] use SVM to learn the facial point response, and then model the global shape using Bayesian objection function. The facial point responses are treated as hidden variable, and they optimize this global model by a consensus of exemplars. Baltrušaitis et al. [13] develop a continuous conditional neural fields (CCNF) method, which defines three types of feature functions, vertex features and two types of edge features to ensure the smooth and sparse constrains. When solving face alignment problem, CCNF generates local response maps, and other global shape models can then predict shapes. Gauss–Newton Deformable Model [14] is another local response based method which jointly optimizes a part-based appearance model along with a global shape model by Gauss–Newton methods. It is effective for the processing during fitting since optimization is done on sparse grid using weighted least-squares.

2.2. Cascaded regression based method

The face alignment task can be modeled by a regression function F, which takes the input image I and predicts the locations of the landmarks θ, i.e. $\theta = F(I)$. And the cascaded regression based method can be formalized as:

$$\theta = F(I) = f_n(f_{n-1}(...(f_1(\theta_0, I), ..., I)I) \quad (1)$$

where $\theta_i = f_i(\theta_{i-1}, I), i = 1, ..., n$ Recently, many cascaded regression based methods have been proposed. Xiong and Torre proposed Supervised Descent Method (SDM) [15], which is a general solution to solve the nonlinear least square problem using supervised descent, and successfully applied it to face alignment tasks. In SDM, f_i is modeled as the linear regression model and the inputs to f_i are the SIFT features related to the shapes. Ren et al. proposed a highly efficient cascaded regression method using efficient Local Binary Features (LBF) [16], which are learned from local regions using random forest. Zhang et al. proposed Coarse to Fine Autoencoders Network (CFAN) [17] in order to learn nonlinear mapping from face image to face shape. CFAN consists of a cascade of multiple stacked autoencoders. Zhu et al. proposed Coarse-to-Fine Shape Searching (CFSS) [18] for face alignment. Unlike other cascaded regression methods, CFSS updates the probability distribution of the sample in the shape space which contains diverse shapes rather than updates shapes directly using regression. Adaptive search over the shape space prevents the results from the sub-optimal solution due to poor initialization. Zhu et al. proposed Cascaded Compositional Learning (CCL) [19] to handle large pose face alignment. CCL first partitions face samples into various domains and learns regressors separately, then combination coefficients are learned to combine predictions on different domains to get final estimation.

2.3. Deep learning based method

Sun et al. [20] trained 23 CNNs cascaded of three levels to do face landmark detection. The first level takes three different face regions as input and predicts the corresponding landmarks locations. The following level predicts the updated landmark location according to the local regions around the location of the previ-

ous level. This is the first work that uses CNNs to conduct feature learning and shape regression for face alignment. Zhu et al. proposed a 3D face shape based face alignment method 3DDFA [18]. RGB image and PNCC (Projected Normalized Coordinate Code) are used to predict the parameters of the 3D face model, which can be used to estimate the face shape. Multiple CNNs are cascaded to get a finer estimation. PNCC is updated according to the predicted face shape and fed as input to the next level. With the 3D face model, a large pose shape can be estimated. Lai et al. [21] proposed to use Fully Convolutional Network to do rough face landmark detection in a segmentation way, with landmark labels encoded as k(# landmarks) Gaussian heatmaps and then used the shape estimated by FCN to train a SDM-like cascaded regression model by Shape-Indexed Pooling features. Zhang et al. proposed Tasks-Constrained Deep Convolutional Network (TCDCN) [22] to do face landmark detection. TCDCN first learns robust face representation from face data with sparse landmark annotations and multiple facial attribute labels, i.e. gender, w/o glasses, poses. Then, a dense landmark model is fine tuned from learned representation using data annotated with dense landmarks. Although cascaded regression is efficient, and each level of the cascade is trained separately, it is not an end-to-end system. Trigeorgis et al. [23] proposed Mnemonic Descent Method (MDM), which used convolutional recurrent neural network to do end-to-end cascaded regression. In this work, expressive representation is learned by CNN and gradients are propagated to each level by RNN. MDM gained significantly better performance than traditional cascaded regression methods.

3. Face alignment by enhanced convolutional neural network

In this section, we introduce our method step by step. First we define the response mapping and the loss function for our deep learning architecture. Second, we go to the details of our enhanced convolutional neural network, and explain our architecture design patterns. Finally, we show how we get the shape from response maps.

3.1. Reconstructing response maps

The objective of enhanced CNN is to learn response maps for given input face image. To define a response map, we make the following assumptions:

(1) The response of the exact interested location (x^*, y^*) is highest.
(2) The farther a point (x, y) is away from (x^*, y^*), the lower the response is.
(3) The sum of responses in each response map equals 1,[1] so it forms a "probability distribution"-like response map.

It is reasonable for us to make these assumptions. Specifically, we choose the probability distribution to be two dimensional Gaussian distribution (and will be normalized to ensure the third assumption).

$$L_k^i \sim \mathcal{N}\left((x_k^{i*}, y_k^{i*}), \sigma^2\right) \tag{2}$$

where L_k^i is the target distribution for the kth feature point of the i training image, (x_k^{i*}, y_k^{i*}) is the corresponding labeled coordinates of the feature point, and σ is the standard deviation to determine the width of the Gaussian window.

The objective of ECNN is to predict the target distribution L. We denote the output of ECNN as $\tilde{C}$, then normalize it to form a probability C. The normalization is conducted by dividing the output of each location by the sum of the outputs of the whole map. This

[1] This is achieved by applying the softmax function in each response map.

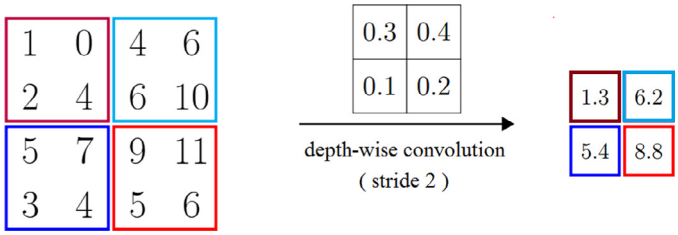


Fig. 2. Illustration of depth-wise convolution operation with stride 2. Depth-wise convolution assigns a weight to each node and then accumulates them. The weights are shared in the same channel. Similar to the commonly used max pooling, the depth-wise convolution reduces the width and height of the feature maps by a half with stride 2.

can be achieved by applying softmax function on $\tilde{C}_i$. Since L_i and C_i are both probability distributions, we choose the loss function to be KL divergence loss:

$$\min \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{j,c} L_k^i(j,c) \log \frac{L_k^i(j,c)}{C_k^i(j,c)} \tag{3}$$

where N is the number of training images, K is the number of response maps for each image, and j and c are the two dimensional coordinates of the feature maps.

The KL divergence loss is actually the same as cross entropy loss, and the residual of these two losses is constant. Another choice of loss function is Euclidean distance between L_i and C_i. We prefer the former one because the plateaus are less present in KL divergence loss than in Euclidean loss [24]. Besides, most elements of L_i and C_i are close to zero, which makes the network using Euclidean loss easily stuck in bad local minima.

3.2. Enhanced convolutional neural network

Convolutional neural network (CNN) has been applied broadly on vision tasks, and shows great success on image classification [25], object detection [26], face recognition [27] and so on. The vanilla CNN is composed of convolutional layers, pooling layers and fully connected layers. The convolutional layers are to extract dense and local feature maps. The pooling layers, typically max pooling layers can reduce dimension, and more importantly, make the feature maps invariant to small deformation. The fully connected layers accumulate all feature maps to predict the final output. The vanilla CNN is suitable for classification problem. For face alignment problem, we aim at reconstructing response map for input face, thus we need to revise vanilla CNN.

To make the CNN adapt to the alignment task, we replace the common used max pooling with *depth-wise convolution* [28]. The max pooling is to return the maximum value of an area, while depth-wise convolution is to assign a weight to each of this area and then accumulate them (see Fig. 2), in which the weight is shared in the same feature map. The depth-wise convolution is applied on a single input channel, so the number of output channels is the same as the input. The depth-wise convolution is better than max pooling for two reasons. One is that max pooling is invariant to small translations like shift and rotation, which is suitable for classification problems. Unfortunately, face alignment problem is translation sensitive, which means that, if we shift, scale or rotate the input face, the outcome should be changed accordingly. Thus, the commonly used max pooling is not suitable for face alignment. The other factor is that depth-wise convolution is parametrized, and the kernel is learned from data, thus it will increase the representation power of the max pooling based model.

One way to boost the performance of CNN is increasing depth. Inspired by works of [29,30], we designed multiple networks, and

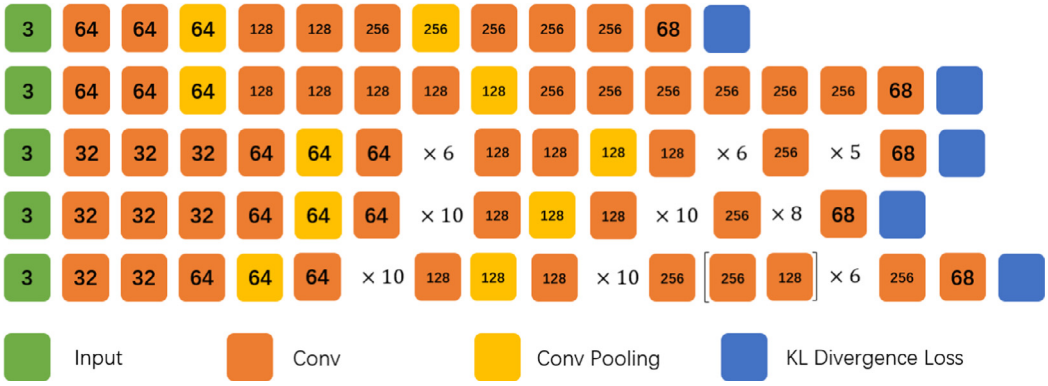
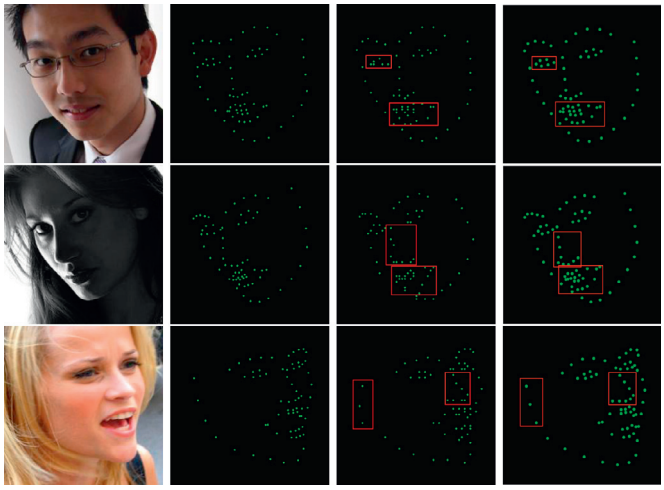


Fig. 3. Architecture of our ECNN models. The depths from the top line to bottom line are {11, 16, 26, 36, 41}. Different colors of blocks mean different layers, and the number in the block means the output channels of the layer. All the filter sizes are set to be 3 with padding 1 to ensure output spatial size not changed. The symbol '×' means the number of reduplicative layers with identical setting. All convolutional (pooling) layers except the last one are followed by ReLU neuron layer. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

their depths are {11, 16, 26, 36, 41}. We find that the deeper network produces the better localization performance. Fig. 3 illustrates the architectures of our ECNN. For face alignment problem, increasing the depth has special meanings. To make discussion clear, we first define the receptive field size (rfs) for a node is that the size of area from the input which would influence its activation. The rfs in the last layer of our model increases as the depth increases. For deeper models, the neurons in the last layer can make a decision from a larger-size image area. That's also a reason why deep models are better for face alignment.

It is hard to train a deep model due to the notorious problem of vanishing and exploding gradients [24]. Several methods are developed to solve this problem, such as robust initialization [24,31], batch normalization [32], and deep residual learning [30]. We use two methods to train our deep model. Due to the reason that we increase the depth of our model gradually, we initialize the deep models using the last learned model. And the first one was initialized randomly. The size of parameters is not coincident in some layers, thus we use a similar way of Net2Net operation [33] to boost the convergence speed and performance of deep models. We also use the gradient clipping strategy [34] which can be regarded as an adaptive learning rate method to solve the gradient exploding.



(a) Input (b) Label (c) *Max* (d) AE

Fig. 4. Effects of auto-encoder model. The shape from *max* operation looks noisy (like the noise and mouth part). The shape is refined by the auto-encoder that characterizes the global shape constraint. (Zoom in to see it clearly.)

3.3. Shape mapping

In shape mapping stage, we get the predicted shape which gives response mappings. There are two simple operations to complete this task, one is to get the location of the max response for each point, the other is to compute the expectation locations from the response maps.

Through the above operations, we can get a good facial point prediction for input face. But the generated shapes contain some outliers, as shown in Fig. 4. These outliers come from two factors. The first is that we do not incorporate the global shape constraints into our ECNN model explicitly, thus some points without clear edge or corner, such as the contour points, are hard to localize. The second is the quantization noise due to pooling operation of our model, because the width of response mapping is only 1/4 of the original input image.

3.4. Auto-encoder based shape rectification

In face alignment, some feature points without clear edge or corner like the contour of the face are hard to localize by the shape mapping of ECNN. Therefore, we build an auto-encoder model to rectify these outlier points by the global shape constraint of the human face. Our autoencoder consists of an encoder with layer 136-90-50 and a symmetric decoder. All units are logistic except for the 50 linear units in the code layer. We use the euclidean loss for reconstruction. Adam optimizer, which is a variation of SGD, is used to solve this problem. As the hidden layer is non-linear, the auto-encoder behaves differently from PCA, with the ability to capture multi-modal aspects of the input distribution. Note that our purpose is to rectify the outlier points of the ECNN output. In order to force the hidden layer to discover more robust shape constraint and prevent it from simply learning the identity, we train the autoencoder to reconstruct the labeled face shape from a corrupted version [35], i.e. the maximum or expectation index from the response maps of ECNN. This methodology is known as denoising autoencoder [35] in the literature.

For comparison purpose, we also implement a PCA shape model to post-process the shape vector, which is inspired by ASM [36]. In the PCA shape model, we first align the train shapes and test shapes, and then apply PCA to extract the main components when training and project input shape into the PCA subspace then reconstruct the shape when testing.

4. Experiments

In this section, we first describe the experimental settings and evaluation criteria. Second, we discuss about the effectiveness of our architecture including the depth and convolution pooling. Third, we discuss different shape mapping operations and give their performances. We also compare our methods with state-of-the-art face alignment algorithms.

4.1. Datasets

We evaluate our proposed methods on the challenging 300-W dataset [11]. The 300-W dataset consists of 3148 training faces collected from HELEN dataset [37], LFPW dataset [12] and AFW dataset [38]. The testing dataset consists of totally 689 faces coming from LFPW, HELEN and the challenging IBUG subset. Evaluations are conducted on three parts: common subset containing 554 images from LFPW and HELEN datasets, challenging subset containing 135 images from IBUG, and the full set of both challenging and the common subsets. We mainly evaluate the 68 points, but our algorithm can be readily extended to extract other numbers of points easily.

4.1.1. Evaluation criteria

We use two commonly used criteria [15,39] to evaluate our algorithm, i.e. mean error rate and cumulative error distribution. The mean error is measured by the distances between estimated landmarks and the ground truths, and normalized with respect to the inter-ocular distance, which makes the criterion invariant to the face size. Specifically, the mean error rate is computed as:

$$e = \frac{1}{N}\sum_{i=1}^{N}\frac{||s_i - s_i^*||_2}{68 * D_i} \tag{4}$$

where N is the number of test shapes, D_i is the distance between the centers of two eyes computed from label shape s^*. The cumulative error distribution (CED), which will treat samples with larger mean error as a failure:

$$\mathrm{CED}(\epsilon) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}\left(\frac{||s_i - s_i^*||_2}{68 * D_i} \le \epsilon\right) \tag{5}$$

where ϵ is the threshold variable and $\mathbb{1}(\cdot)$ is the indicator function. The CED is typically illustrated as a curve to describe the proportion of the test images whose error rates are smaller than a certain value, which displays more comprehensive performance than the mean error rate.

4.1.2. Data augmentation

We train our model from scratch without external data. To alleviate over-fitting, we expand the dataset by doing scale and rotation transformation. We first crop the face with bounding box enlarged by $\{1.2\times, 1.3\times, 1.4\times\}$, and for each scale, we rotate it with angle $\{\pm 30°, \pm 25°, \pm 20°, \pm 15°, \pm 10°, \pm 5°, 0°\}$.

4.2. Training enhanced convolutional neural network

4.2.1. Implementation details

We implement our enhanced convolutional neural network using deep learning toolbox Caffe [40]. We train our ECNN model by stochastic gradient descent (SGD) with momentum 0.9 and learning rate 0.02. All the faces are resized to 224×224. The mini-batch size is set to be 32 and we use L2 regularization with the decay parameter of 0.0001. We increase the depth from 11 to 41 gradually. The first ECNN is trained from scratch, while other deeper ECNNs are initialized from the model of last CNN to boost the training speed and to get a better result. Deep neural networks are very hard to train, so that we use the gradient clipping technique described in [34], and clip the length of gradient to 10. We do not tune these parameters much, and stop tuning when we get a reasonable convergence. We believe that carefully tuning these hyper-parameters will get more precise predictions, but it is not our focus. Table 1 lists the basic results done on 300-W dataset.

Table 1
The receptive field size (rfs), average forward time (in millisecond), and localization accuracy on 300-W dataset with ECNNs of different depths.

Depth	rfs	Time (ms)	Mean error (full/challenge/common)	
			Max shape mapping	Mean shape mapping
11	58	17.5	10.57/18.16/8.71	8.91/16.26/7.12
16	88	20.7	7.57/12.84/6.28	6.39/11.29/5.19
26	146	22.3	6.77/10.91/5.76	5.86/9.85/4.89
36	212	25.1	6.21/10.16/5.25	**5.46/9.27/4.53**
41	260	26.8	6.23/10.24/5.25	5.50/9.33/4.57

Table 2
Comparison between depth-wise convolution and max pooling.

Depth	Mean error (full/challenge/common)		
	Max pooling str 2	Depth-wise conv str 2	Depth-wise conv str 1
11	9.77/17.96/7.78	8.66/15.87/6.91	7.36/13.48/5.87
16	6.49/11.39/5.30	6.28/11.08/5.12	6.02/10.63/4.91
26	5.88/9.69/4.95	5.78/9.67/4.83	5.66/9.47/4.73
36	5.57/9.15/4.69	**5.46/9.27/4.53**	5.51/9.36/4.57

4.2.2. Depth-wise convolution vs. max pooling

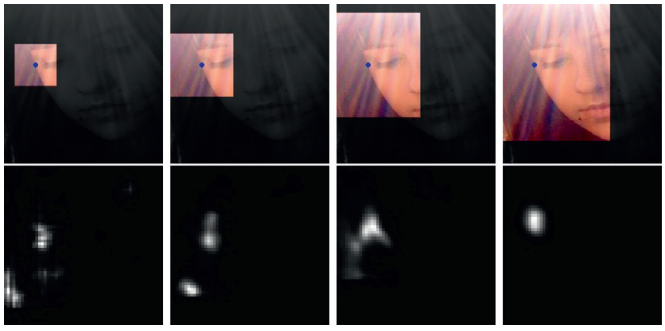
Table 2 shows performance comparison between depth-wise convolution and max pooling. We can see that replacing max pooling with depth-wise convolution will decrease the error rate by a large degree, which validates the ability of depth-wise convolution for alignment problem. To explore whether the depth-wise convolution of stride 2 deteriorates the localization accuracy, we also implement the CNNs with the depth-wise convolution layers of stride 1 to keep the size of the feature maps as the input image.

As emulated in Table 2, for the relatively shallow CNNs of 11 layers and 16 layers, the localization accuracy improves significantly by replacing stride 2 with stride 1 without decreasing the feature map size. However, as the CNN becomes deeper, the advantage of keeping the size of feature maps becomes trivial. This may be that the convolution on the reduced size feature maps related to the larger receptive field of the original image. In addition, our mean shape mapping method is also robust to the quantization noise caused by the reduced feature maps.

Note that decreasing the feature mapping sizes can largely reduces the computational load of the deep convolution operators. Since the size of the feature maps is reduced to 1/4 and 1/16 after the first and second depth-wise convolutional layer of stride 2, the computational load on the convolution operators can be largely reduced. With the comparable accuracy, we therefore prefer to apply the depth-wise convolution with stride 2 in our ECNN method for the computational efficiency purpose.

4.2.3. Depth, receptive fields and performance

As we increase the depth of our model, the mean error is decreased rapidly from 8.66% to 5.43% for depth-wise convolution, which shows deep models are far more powerful than shallow ones. Fig. 5 shows the rfs and response maps, which suggests that, as rfs increases, the model can predict a more accurate response map from a larger context area. The performance improvement is saturated when the rfs reaches the image size, continuing increasing the depth and rfs no longer yields lower error rate, as shown



(a) rfs=58 (b) rfs=88 (c) rfs=146 (d) rfs=212

Fig. 5. Illustration of receptive field size (first row) and response map for the first key point (second row). As rfs increases, the model can see more area then make a decision, and thus will get more accurate response map.

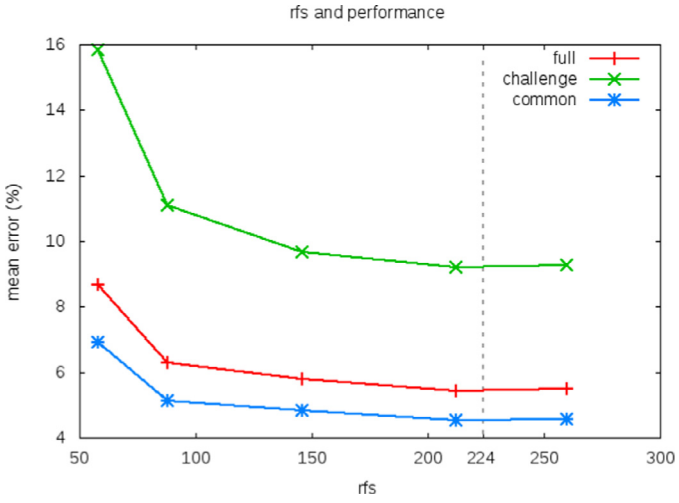


Fig. 6. The figure of rfs and performance. The gray dashed line shows the image size. The localization error decreases as the rfs increases.

in Fig. 6. We test the ECNN on Ubuntu 16.04, Nvidia TITAN X, on which the average forward speed of ECNN with different depths is enumerated in Table 1.

4.3. Shape mapping

4.3.1. Max vs. mean shape mapping

The mean shape mapping method is always better than max shape mapping method. Our 36 layer model gives 6.21% mean error for max shape mapping, while 5.46% for for mean shape mapping. The reasons come from two factors. The first is that max pooling only predicts integral indexes. This will add quantization error on its performance. The other is that mean pooling uses more information in the response map to give a prediction. Since we initially reconstruct a Gaussian response mapping, the mean shape mapping can be seen as approximating a Gaussian distribution on predicted response map and then returning its mean.

4.3.2. Shape rectification

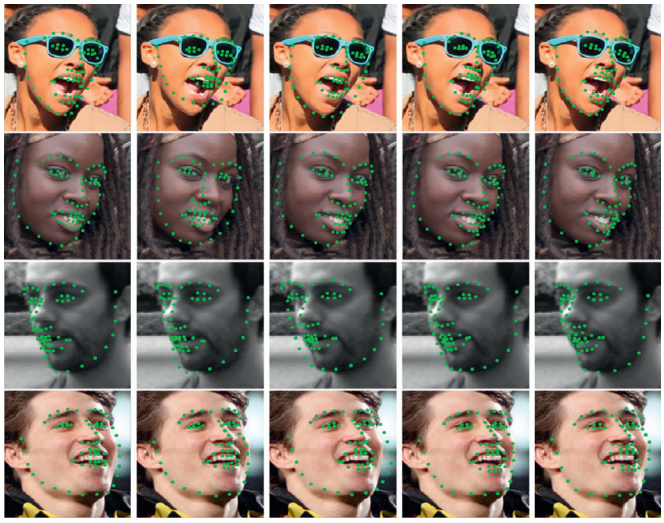
We treat the auto-encoder model as noise reduction function. The noise comes from the implicit encoding of global shape from our PCN model and the quantization noise from max operation. From Table 1, auto-encoder model takes effects on max and mean shape mapping. And it takes more effect on max operation than on mean operation, it is reasonable since max operation generates more noise due to quantization error. And we can also observe that

Table 3
Performance of autoencoder on 300-W dataset with response maps generated from 36 layer ECNN.

Methods	Full	Challenging	Common
Max	6.21	10.16	5.25
Max + PCA	5.71	9.58	4.77
Max + Autoencoder	5.57	9.66	4.57
Mean	5.46	9.27	4.53
Mean + PCA	5.43	9.19	4.52
Mean + Autoencoder	**5.32**	**9.18**	**4.38**

Table 4
Mean error on 300-W datasets.

Methods	Full	Common	Challenging
SDM [15]	7.52	5.60	15.4
RCPR [41]	8.35	6.18	17.26
CFAN [17]	7.69	5.50	16.78
ESR [39]	7.58	5.28	17.00
LBF [16]	6.32	4.95	11.98
CFSS [18]	5.76	4.73	9.98
ECNN(ours)	**5.32**	**4.38**	**9.18**



(a) SDM (b) CFAN (c) CFSS (d) Ours (e) Label

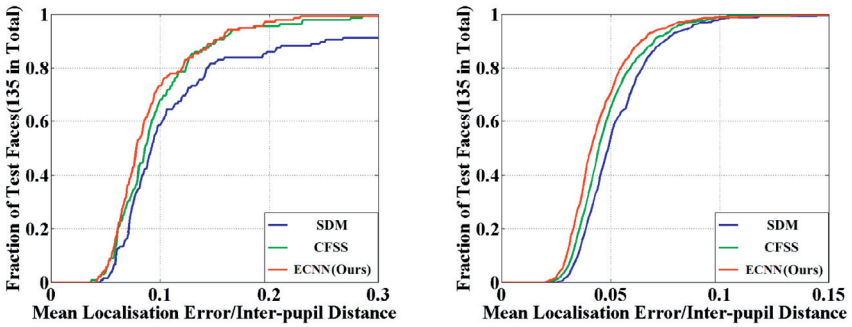
Fig. 7. Examples of SDM [15], CFAN [17], CFSS [18]. The images are from challenging IBUG subset from 300-W dataset.

the auto-encoder is far more better than PCA noise reduction models, which shows the deep non-linear models are superior to linear models.

Comparing the results in Table 3 with the results of pure ECNN in Table 2, we find that the ECNN + autoencoder method indeed yields better accuracy. This result convinces us that the autoencoder increases the quality of the results by incorporating the global shape constraints into our ECNN model explicitly.

4.4. Compared with other methods

To illustrate the performance of our model, we make a comparison with other methods. The overall results are enumerated in Table 4, in which we report our results of our 36 layer model and *mean + AE* shape mapping. We achieve the mean error of 5.32%, outperforming the state-of-the-art 5.76% from CFSS [18]. Fig. 7 shows some alignment examples of SDM [15], CFAN [17], CFSS [18] and our method. As one can see from the figures, the proposed method locates the feature points more accurately, es-



(a) Challenging subset (b) Common subset

Fig. 8. Comparisons of cumulative error distributions (CED) curves.

pecially for the face contours with self-occlusion. This indicates that the auto-encoder is effective for the shape rectification by the global shape constraint. The comparative CED curves are illustrated in Fig. 8, which suggests that the proposed ECNN method consistently outperforms the SDM and CFSS methods. Table 4 enumerates the comparative mean error rates of six state-of-the-art methods on the 300-W dataset, which clearly shows the superior performance of the proposed method on all the three subsets.

5. Conclusions and future works

In this work, we solve the face alignment problem by two stages. In the response mapping stage, we reconstruct the response maps for each key point for given face image. In the shape mapping stage, we predict the shape based on response maps. With *mean + AE* operation, our model achieves the mean error of 4.38% on the common subset of 300-W dataset. Besides, our model is theoretically shift-invariant, which is important in real application.

In the future, we will incorporate the global shape constraints into the ECNN architecture to fully utilize the power of deep models. Besides, the ECNN can also be applied to other location related tasks, like face detection.

Acknowledgments

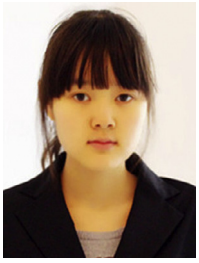
This work was partially supported by Beijing Nova Program under grant no. Z161100004916088, National Natural Science Foundation of China (projects 61573068, 61471048, and 61375031), and the Fundamental Research Funds for the Central Universities under grant no. 2014ZD03-01.

References

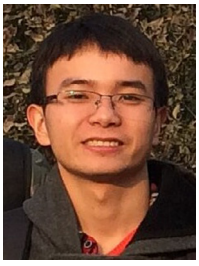
[1] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld, Face recognition: a literature survey, ACM Comput. Surv. 35 (4) (2003) 399–458.
[2] E. Murphy-Chutorian, M.M. Trivedi, Head pose estimation in computer vision: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 31 (4) (2009) 607–626.
[3] V. Blanz, T. Vetter, Face recognition based on fitting a 3D morphable model, IEEE Trans. Pattern Anal. Mach. Intell. 25 (9) (2003) 1063–1074.
[4] L. Zafeiriou, E. Antonakos, S. Zafeiriou, M. Pantic, Joint unsupervised face alignment and behaviour analysis, in: Computer Vision–ECCV 2014, Springer, 2014, pp. 167–183.
[5] P. Campadelli, G. Lipori, R. Lanzarotti, Automatic Facial Feature Extraction for Face Recognition, INTECH Open Access Publisher, 2007.
[6] P. Viola, M.J. Jones, Robust real-time face detection, Int. J. Comput. Vision 57 (2) (2004) 137–154.
[7] M.C. Burl, T.K. Leung, P. Perona, Face localization via shape statistics, in: International Workshop on Automatic Face and Gesture Recognition, 1995, pp. 154–159.
[8] D. Vukadinovic, M. Pantic, Fully automatic facial feature point detection using Gabor feature based boosted classifiers, in: 2005 IEEE International Conference on Systems, Man and Cybernetics, 2, IEEE, 2005, pp. 1692–1698.
[9] D. Cristinacce, T.F. Cootes, I.M. Scott, A multi-stage approach to facial feature detection, in: BMVC, 2004, pp. 1–10.
[10] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.
[11] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, M. Pantic, 300 faces in-the-wild challenge: the first facial landmark localization challenge, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013, pp. 397–403.
[12] P.N. Belhumeur, D.W. Jacobs, D.J. Kriegman, N. Kumar, Localizing parts of faces using a consensus of exemplars, IEEE Trans. Pattern Anal. Mach. Intell. 35 (12) (2013) 2930–2940.
[13] T. Baltrušaitis, P. Robinson, L.-P. Morency, Continuous conditional neural fields for structured regression, in: European Conference on Computer Vision, Springer, 2014, pp. 593–608.
[14] G. Tzimiropoulos, M. Pantic, Gauss-newton deformable part models for face alignment in-the-wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1851–1858.
[15] X. Xiong, F. Torre, Supervised descent method and its applications to face alignment, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 532–539.
[16] S. Ren, X. Cao, Y. Wei, J. Sun, Face alignment at 3000 fps via regressing local binary features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1685–1692.
[17] J. Zhang, S. Shan, M. Kan, X. Chen, Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment, in: Computer Vision–ECCV 2014, Springer, 2014, pp. 1–16.
[18] S. Zhu, C. Li, C. Change Loy, X. Tang, Face alignment by coarse-to-fine shape searching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4998–5006.
[19] S. Zhu, C. Li, C.C. Loy, et al., Unconstrained face alignment via cascaded compositional learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3409–3417.
[20] Y. Sun, X. Wang, X. Tang, Deep convolutional network cascade for facial point detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3476–3483.
[21] H. Lai, S. Xiao, Z. Cui, Y. Pan, C. Xu, S. Yan, Deep cascaded regression for face alignment, arXiv:1510.09083 (2015).
[22] Z. Zhang, P. Luo, C.C. Loy, X. Tang, Learning deep representation for face alignment with auxiliary attributes, IEEE Trans. Pattern Anal. Mach. Intell. 38 (5) (2016) 918–930.
[23] G. Trigeorgis, P. Snape, M.A. Nicolaou, E. Antonakos, S. Zafeiriou, Mnemonic descent method: a recurrent process applied for end-to-end face alignment, in: Proceedings of IEEE International Conference on Computer Vision & Pattern Recognition (CVPR16), Las Vegas, NV, USA, 2016.
[24] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.
[25] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[26] R. Girshick, Fast R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[27] Y. Sun, Y. Chen, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, in: Advances in Neural Information Processing Systems, 2014, pp. 1988–1996.
[28] F. Chollet, Xception: deep learning with depthwise separable convolutions, arXiv preprint, arXiv:1610.02357(2016).
[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[30] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, arXiv preprint, arXiv:1512.03385(2015a).
[31] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on ImageNet classification, in: Proceedings of the IEEE International Conference on Computer Vision, 2015b, pp. 1026–1034.

[32] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proceedings of The 32nd International Conference on Machine Learning, 2015, pp. 448–456.
[33] T. Chen, I. Goodfellow, J. Shlens, Net2net: accelerating learning via knowledge transfer, arXiv preprint, arXiv:1511.05641(2015).
[34] R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, in: Proceedings of the 30th International Conference on Machine Learning (ICML-13), 2013, pp. 1310–1318.
[35] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408.
[36] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, Active shape models-their training and application, Comput. Vision Image Understanding 61 (1) (1995) 38–59.
[37] V. Le, J. Brandt, Z. Lin, L. Bourdev, T.S. Huang, Interactive facial feature localization, in: Computer Vision–ECCV 2012, Springer, 2012, pp. 679–692.
[38] X. Zhu, D. Ramanan, Face detection, pose estimation, and landmark localization in the wild, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 2879–2886.
[39] X. Cao, Y. Wei, F. Wen, J. Sun, Face alignment by explicit shape regression, Int. J. Comput. Vision 107 (2) (2014) 177–190.
[40] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: Proceedings of the ACM International Conference on Multimedia, ACM, 2014, pp. 675–678.
[41] X. Burgos-Artizzu, P. Perona, P. Dollár, Robust face landmark estimation under occlusion, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1513–1520.

Weihong Deng received the B.E. degree in information engineering and the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2004 and 2009, respectively. From October 2007 to December 2008, he was a postgraduate exchange student in the School of Information Technologies, University of Sydney, Australia. He is currently an associate professor in School of Information and Telecommunications Engineering, BUPT. His research interests include statistical pattern recognition and computer vision, with a particular emphasis in face recognition. He has published over 90 technical papers in international journals and conferences, such as IEEE TPAMI and CVPR. He serves as guest editor for Image and Vision Computing journal and the reviewer for several international journals, such as IEEE TPAMI/TIP/TIFS/TNNLS/TMM/TSMC, IJCV, PR/PRL. Recently, he gives tutorials on face recognition at ICME 2014, ACCV 2014, CVPR 2015 and FG 2015, and organizes the workshop on feature and similarity learning in ACCV2014 with colleagues. His dissertation titled “Highly accurate face recognition algorithms” was awarded the Outstanding Doctoral Dissertation Award by Beijing Municipal Commission of Education in 2011. He has been supported by the program for New Century Excellent Talents by the Ministry of Education of China in 2013 and Beijing Nova Program in 2016.



Yuke Fang is a junior student studied in International School of Beijing University of Posts and Telecommunications (BUPT). She majors in Telecommunications Engineering with Management and will graduate in June 2018, whose interests are in computer vision and pattern recognition.



Zhenqi Xu received the B.E. degree in telecommunication engineering from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2014. He currently is a post-graduate student major in Information and Telecommunications Engineering. His research interests include pose-invariant face recognition and deep learning.

Jiani Hu received the B.E. degree in telecommunication engineering from China University of Geosciences in 2003, and the Ph.D. degree in signal and information processing from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2008. She is currently a lecturer in School of Information and Telecommunications Engineering, BUPT. Her research interests include information retrieval, statistical pattern recognition and computer vision.