FACIAL LANDMARK DETECTION VIA CASCADE MULTI-CHANNEL CONVOLUTIONAL NEURAL NETWORK

Qiqi Hou, Jinjun Wang, Lele Cheng, Yihong Gong

Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
28 Xianning West Road, Xi'an, Shaanxi, China
{houqiqi, ch_lele2012}@stu.xjtu.edu.cn, {jinjun, ygong}@mail.xjtu.edu.cn

ABSTRACT

This paper presents a novel cascade multi-channel convolutional neural networks(CMC-CNN) approach for face alignment. Several CNN are jointly used for the finally output. In our method, each stage CNN takes the local region around the landmarks as input, and each local patches does convolution separately, which can lead network to learn local high-level features. Then a fully connected layer is put to learn global information from these local features. Our methods has achieves the state-of-the-art results when tested on the 300 Face in-the-Wild(300-W) dataset.

***Index Terms*—** Face Alignment, CMC-CNN, Local Feature, Global Feature

1. INTRODUCTION

Facial landmark detection plays an important role in many face analysis tasks, such as facial expression recognition, face verification, and face recognition [1, 2, 3, 4]. The problem has been extensively studied in recent years. For instance, the well-known Active Shape Model [5] and the Active Appearance Model [6] try to fit a generative model for global facial appearance, such as Principal Component Analysis [7]. Although they are robust to local corruptions, parameter estimation for these generation models usually requires expensive iterative steps. Besides, in real-world applications, these approaches usually fail when there exists complex appearance variations.

To handle real-world and complex facial landmark detection scenario, some researchers focused on constructing face templates to fit input images. For example, the recent stacked deformable shape model [8] has achieved promising progress. The limitations of these methods are mainly their huge computational cost.

More recently, the regression-based methods have shown to be very effective for facial landmark localization [10, 11, 12, 13]. Using local patches around selected facial landmarks or over the entire image region, regressors are trained as a predictor to compute the landmark coordinates. These methods are robust and efficient and have obtained the state-of-the-art performance. To list some examples, the explicit shape regression (ESR) [11] used Haar-like feature and random ferns. The supervised descent method (SDM) [14] used SIFT and linear regressor. Face alignment with LBF [13] used local binary features and linear regressors. The coarse-to-fine auto-encoder (CFAN) uses HOG feature and auto-encoder as their regressor at local stages. Most of these methods depends on handy-crafted feature for processing, and therefore it is more desirable to perform local feature learning and regressor training jointly.

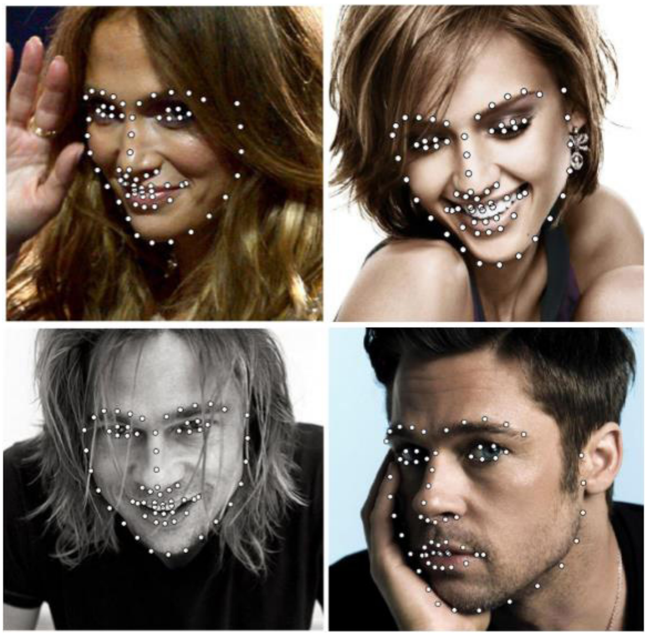

Fig. 1. Selected facial landmark detection result from the 300-W dataset [9]

Some researchers have attempted cascade of deep convolutional neural networks (CNN) models to automate the feature learning process for coarse-to-fine facial landmark de-

Thanks to XYZ agency for funding.

tection [15, 16]. For example, [15] applied a first CNN to get the coarse face location, from which the precise locations of facial landmarks were next detected using a second CNN. Although the method achieved very promising accuracy, the multiple CNN models applied at different stage of the detection process break down the global information as a consistent constraint.

The situation has motivated us to combine both the local feature and global information into the regression framework. In this paper, we design an Cascade Multi-Channel CNN (CMC-CNN) model that is capable of coarse-to-fine facial landmark detection through an cascade process. Unlike that in [15], our model only uses one CNN model in each detection task. As illustrated in Figure 1, our landmark detection process consists of multiple bottom-up detection and top-down correction pairs, such that both the local information and global information could be utilized in a generic framework. Experimental results have shown that our method is accurate and fast.

2. CASCADE MULTI-CHANNEL CNN

In this paper, we present a multi-channel convolutional neural network for facial landmark detection. The model works in an cascade fashion where the initial locations of multiple patches are fed into the bottom-up path of the model to calculate the corrections of landmark coordinates. The top-down path then guesses new landmark coordinates, and such bottom-up/top-down process iterates until convergency. The next subsections elaborate the proposed Cascade Multi-Channel CNN (CMC-CNN) model.

2.1. The cascade of regressors

Let $S \in \mathbb{R}^{2*p}$ be the coordinates of facial landmarks in an image I, where p denote the number of facial landmarks. In this paper, we refer to the vector S as a shape, S^t as the estimate of S at stage t, and $R^t$the regressor at stage t. The ground truth shape is $\hat{S}$.

With N training samples $\{I_i, \hat{S}_i, S_i^0\}_{i=1}^N$, training process wants to reduce the alignment errors on training set. Specifically The t stage regressor is formally learnt as follows

$$R^t = \underset{R^t}{argmin} \sum_{i=1}^{N} \left\| \hat{S}_i - S_i^{t-1} - R^t(I_i, S_i^{t-1}) \right\|_2,$$

where S_i^{t-1} is the estimated shape in the previous stage $t-1$. Note that all shapes in our experiments are normalized by meanshape like ESR[11].

In testing, with a facial image I and an initial shape S^0, we update face shape in a cascade manner:

$$S_i^t = S_i^{t-1} + R^t(I_i, S_i^{t-1}).$$

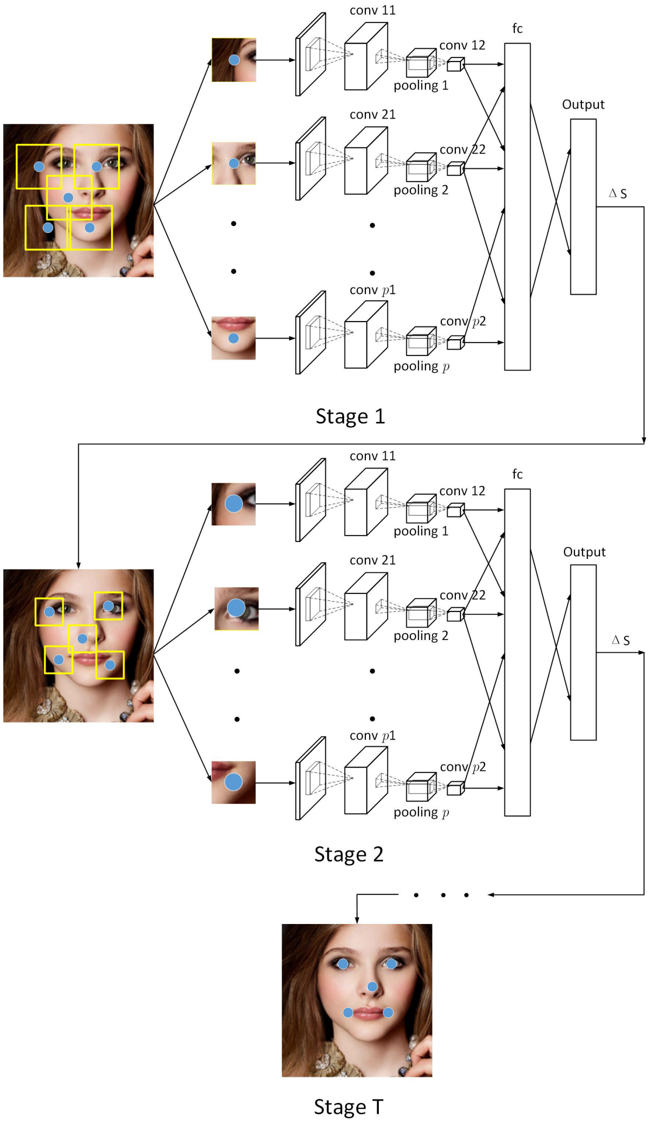

Fig. 2. Overview of our approach. *Given an image and initial shape, CMC-CNN extracts local patches around each landmark. These patches are fed as different channels into our CMC-CNN and then combined through a full connection layer to predict the correction of landmark coordinate, until convergency*

The stage regressor computes ΔS^t based on the previous shape S^{t-1} and image I. In this framework, shape S should be more and more close to the ground truth shape $\hat{S}$ though cascade regressing.

2.2. Multi-Channel CNN

As shown in Figure 1, we take CNN as our regressor. In each stage, we first get the local parch at each landmark, Then these patch are resize to the same size, 15*15 in our experiments.

The network takes the raw pixels as input and performs regression on the location of landmarks. Two convolutional layer are stacked after the input layer. Note that each patches does convolution separately. Finally a fully connected layer connect all local convolutional layers output together.

For convolutional layer, we use Rectified Linear Units (ReLUs) as our neurons. Then convolutional layer can be represented as follow:

$$O_{i,j,k} = max(\sum_{x=0}^{h-1}\sum_{y=0}^{w-1}\sum_{z=0}^{c-1} I_{i-x,j-y,z} \cdot W_{x,y,z,k} + B_k, 0),$$

where I is the input to the convolutional layer, and O is the output. W is the weight and B is the bias.h, w, c denote the width, height and channel of filter. Respectively k means that is the k^{th} filter.

Pooling layers in the network can summarize the output of neighboring groups of neurons in the same kernel map. we used max pooling non-overlapping pooling regions, which can be represent as follows:

$$O_{i,j,k} = \underset{\leq x \leq p, 0 \leq y \leq p}{max} (I_{i \cdot d+x, j \cdot d+y,k}).$$

Instead of combining many different models, Dropout technique is a very effective way to reduce test errors. This technique can reduce complex co-adaptations of neurons. In our test time, we multiply all neurons output by 0.5, which is a reasonable approximation.

For loss layer, we use Euclidean-loss., which computes:

$$loss = \frac{1}{2N}\sum_{i=1}^{N} \left\|O_i - \Delta S_i\right\|_2,$$

where N is the batch-size. O is the network output, ΔS is the regression target, difference between ground truth shape and current shape. Euclidean-loss has many drawbacks, but in this place it is enough. In our future work, we will try other loss.

2.3. Discuss

Differences with LBF. Both LBF and our work consider the pixels in the local region of a landmark and the fully connected layer in our model can be considered as a simplified regressor like that in the LBF, if the local convolutional layer are seen as feature extractors. But there are serval differences: 1) Compared to LBF which employs LBF feature, we take the raw pixels as input. CNN is a strong tool that can learn high-level features itself. At this point, we differ to many other approaches, such as SDM using SIFT feature, CFAN using Hog. 2) LBF employs linear regression to model the mapping from LBF features to a face shape, while our model uses nonlinear regression.

Differences with DCNN. Both DCNN and our model use nonlinear regression and take raw pixels as input. But in DCNN builds at least one CNN for each landmarks, which makes it very hard to process large number of landmarks, such as 68 and 194 in [9, 17]. Besides in DCNN after the first stage, each landmark is refined independently, which limits the model capacity at utilizing global feature, such that the model's accuracy heavily relied on the first stage that uses the global feature.

3. EXPERIMENTS

We conducted experiments on the 300 Face in-the-Wild Challenge dataset (300-W) [9] which is created from several well-known public datasets including LFPW, AFW, Helen, XM2VTS and IBUG. Each face image contains 68 facial landmarks. Following the protocol suggested by [13], our training set consist of the training sets of LFPW and Helen, with 3148 images in total. The testing set has two parts, specifically the common subset and the challenging subset. The former consists of the testing sets of LFPW and Helen, with 544 images in total, while the later is the complete IBUG set with 135 images.

The normalized inter-pupil distance error metric was used to evaluate the landmark detection performance. The error averaged over all landmarks and images is calculated by,

$$error = \frac{1}{N}\sum_{i=1}^{N} \frac{\frac{1}{p}\sum_{j=1}^{p}\left\|S_{i,j} - \hat{S}_{i,j}\right\|^2}{\left\|l_i - r_i\right\|^2},$$

where N is the number of images, p is the number of landmarks in each image (68 in our case), S is the shapes to compute error, $\hat{S}$ is the ground truth, and l and r are the position of the left eye corner and right eye corner, respectively.

3.1. Implementation

To train our model, we augmented the training data by randomly sampling face images with slightly shifted bounding box, each with a flipped version also. In this way we obtained 75,552 images for training, and their mean shape was used for initialization. We used the well-known Caffe [18] to implement a four stage cascade CNN in the experiments, and the parameters of each stage CNN can be found in Table 1. The neural networks are trained by stochastic gradient descent with momentum set to 0.9 and mini-batch size set to 128. We have set an equal learning rate for all learnable layers to 0.01, and it is manually decreased each time by an order of magnitude once the validation error stopped decreasing, to a final rate of 0.0001. For the first stage, each layer's weights are initialized from a zero-mean Gaussian distribution with σ set to 0.01 and biases set to 0. During testing, we used 10 initial shape to get the final location of landmarks, which required 150ms for one image. Figure 3 shows the channel structure for one local patch.

Network	Stage 1	Stage 2	Stage 3	Stage 4
patch ratio	0.3	0.2	0.1	0.1
input	$15^2 \times 1$	$15^2 \times 1$	$15^2 \times 1$	$15^2 \times 1$
conv.1	$5^2 \times 16$	$5^2 \times 16$	$5^2 \times 16$	$5^2 \times 16$
conv.2	$3^2 \times 8$	$3^2 \times 8$	$3^2 \times 8$	$3^2 \times 8$
fc	1024	1024	1024	1024
output	136	136	136	136

Table 1. **Summary of network structures**. Patch ratio denote the ratio between the local patch size and face bounding box. Input, conv.1 and conv.2 is the structure of each local patch and fc is the fully connected layer structure

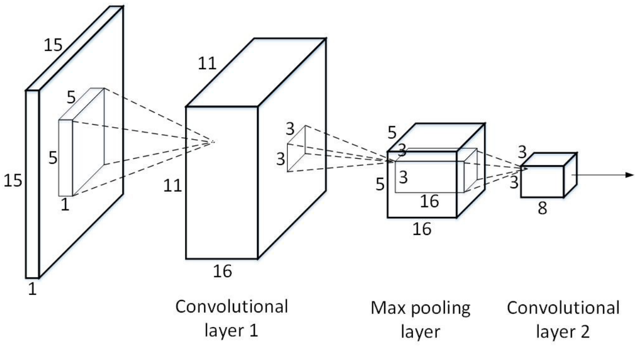

Fig. 3. **Structure of a local patch's channel.** Each channel includes 2 convolutional layers and 1 pooling layers.

Table 2 lists the error at different stages. It is clear that the error reduces through cascade processing.

Output	Error value
Stage 1	6.73
Stage 2	5.18
Stage 3	4.93
Stage 4	4.91

Table 2. **Error at different stages.** The common subset test error at each stage. The error is inter-pupil distance normalized landmark error averaged over all landmarks and images

3.2. Performance

We compared our methods with three existing approaches, specifically the explicit shape regression (ESR) [11], the supervised descent model(SDM) [14] and the local binary features (LBF) [13]. As can be seen from Table 3, our method outperformed these method by incorporating both the local feature in a bottom-up process and the global constraint in a top-down process into a generic regression framework. Figure 4 shows some example image results.

Method	Full	Common	Challenging
ESR	7.58	5.28	17.00
SDM	7.52	5.60	15.40
LBF	6.32	4.95	**11.98**
LBF fast	7.37	5.38	15.50
Our method	**6.30**	**4.91**	12.03

Table 3. **Comparison of facial landmark with state-of-the-art methods.** All the error we used is the original results in the literature in LBF.

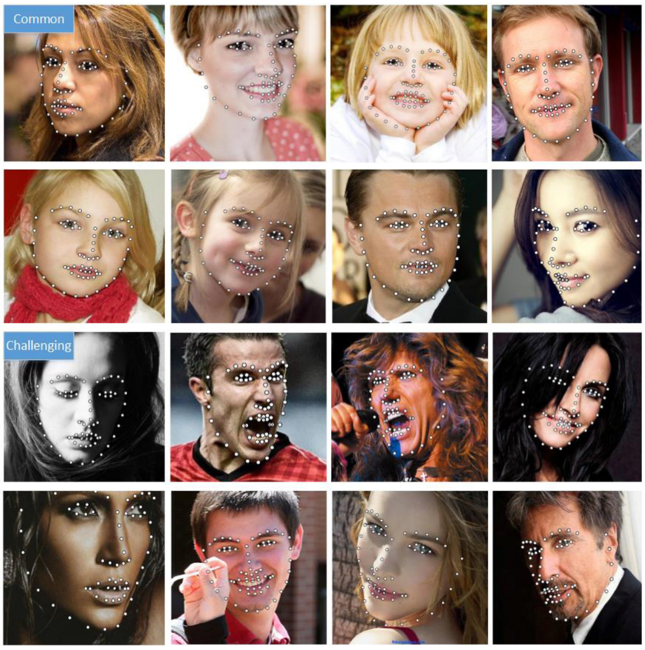

Fig. 4. Example results from the 300-W dataset

4. CONCLUSION

This paper presents a cascade multi-channel CNN model that is capable of performing coarse-to-fine facial landmark detection through cascade of bottom-up detection and top-down correction. Both the local feature information and global prior constraint could be effectively utilized into the detection process to obtain accurate facial landmark coordinates. The proposed model takes raw pixels as input and is very efficient. Our experiments show that the model achieves the state-of-the-art results on the 300-W dataset. The next step of the research is to further consider the face detection task jointly with the landmark detection process through a multi-task learning framework, such that faces could be reliably detected and aligned from complex real-world environment. The work is on-going.

5. REFERENCES

[1] Brian Amberg and Thomas Vetter, "Optimal landmark detection using shape models and branch and bound," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 455–462.

[2] Dong Chen, Xudong Cao, Fang Wen, and Jian Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 3025–3032.

[3] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1701–1708.

[4] Xiaowei Zhao, Tae-Kyun Kim, and Wenhan Luo, "Unified face analysis by iterative multi-output random forests," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1765–1772.

[5] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham, "Active shape models-their training and application," *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[6] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor, "Active appearance models," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[7] Ian Jolliffe, *Principal component analysis*, Wiley Online Library, 2005.

[8] Junjie Yan, Zhen Lei, Yang Yang, and Stan Z Li, "Stacked deformable part model with shape regression for object part localization," in *Computer Vision–ECCV 2014*, pp. 568–583. Springer, 2014.

[9] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic, "A semi-automatic methodology for facial landmark annotation," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*. IEEE, 2013, pp. 896–903.

[10] Piotr Dollár, Peter Welinder, and Pietro Perona, "Cascaded pose regression," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1078–1085.

[11] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun, "Face alignment by explicit shape regression," *International Journal of Computer Vision*, vol. 107, no. 2, pp. 177–190, 2014.

[12] Jie Zhang, Shiguang Shan, Meina Kan, and Xilin Chen, "Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment," in *Computer Vision–ECCV 2014*, pp. 1–16. Springer, 2014.

[13] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun, "Face alignment at 3000 fps via regressing local binary features," .

[14] Xuehan Xiong and Fernando De la Torre, "Supervised descent method and its applications to face alignment," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 532–539.

[15] Yi Sun, Xiaogang Wang, and Xiaoou Tang, "Deep convolutional network cascade for facial point detection," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 3476–3483.

[16] Erjin Zhou, Haoqiang Fan, Zhimin Cao, Yuning Jiang, and Qi Yin, "Extensive facial landmark localization with coarse-to-fine convolutional network cascade," in *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*. IEEE, 2013, pp. 386–391.

[17] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir Bourdev, and Thomas S Huang, "Interactive facial feature localization," in *Computer Vision–ECCV 2012*, pp. 679–692. Springer, 2012.

[18] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.