68 Face Feature Points Detection Based on Cascading Convolutional Neural Network with Small Filter

Yinhuan Zheng [1, a, *], Beizhan Wang [2, b], Yilong Zheng [1]

[1] Computer and AI College, Xiamen Institute of Technology, Amoy China

[2] Software Engineer department, Xiamen University, Amoy China

[*, a] Corresponding author e-mail: yhzheng1218@gmail.com, [b] bzwang@xmu.edu.cn

Abstract. Facial detection has received more and more attention in the past two decades. Due to the pose, occlusion, and illumination changes in the photo, the detection task is quite a challenge in an unconstrained environment. This paper proposed a cascaded convolutional neural network DCNNSF-CFC (Deep Convolution Neural Network with Small Filter-with Coarse-to-Fine Cascade) to localize large facial landmarks to improve the accuracy and robustness of network prediction which based on the original small filter deep convolutional neural network. Each network is trained separately to locally refine a subset of facial landmarks generated at the previous network level, and each geometric network is explicitly constrained to modify the input information of the current network. AFW, LFPW and Helen datasets were used in this study and it was proved to be more accurate and robust than DRMF, Opm, CNN and RCPR.

Keywords: Cascade convolution, facial feature point detection, small filter.

1. Introduction

Recently, cascade regression has become one of the most popular and advanced methods for face alignment because of its high accuracy and high speed [1, 2, 3, 4, 5]. Generally, it's a great challenge to perform regression from image features to face shape in one stage, but the regression process can be performed in stages by learning cascading vector regression. Sun et al. [2] brought the idea of cascade regression into a convolutional neural network, and proposed a cascade convolutional neural network. They carefully designed a three-level convolutional network to handle face alignment tasks, and fused the output of multiple networks at each level for robust prediction. First, the entire face image is used as an input to predict the initial estimation of the overall face shape. Then determine the rough location of each feature point, and then output it to the next level network to achieve higher accuracy and then achieve rough-to-fine effect. The main problem can be briefly described as follows: Enter a picture and get 68 overall facial feature points, including 51 inner facial features including the eyes, eyebrows, mouth and nose, and 17 other contour points.

In order to solve this problem, this paper divides the face into two parts: the outer contour point and the inner feature point basing on[6], further refines the inner feature points of the face ,and then leverage geometric constraints' global layout of facial components and the interaction of feature points in each facial component of the face, finally designs a multilayer cascade convolutional neural network as below:

Level1: learn and predict the whole face, using the DCNNSF [7] to locate the initial 68 human face feature points, taking full advantage of the geometric constraints of the global arrangement of the face components;

Level2: Predict the boundary of the inner feature point and the outer contour point, respectively, and their relative position, reducing the amount of single picture input and predicting the number of feature points to improve accuracy and robustness.

Level3: Reclassify the inside information of the face and predict it separately into six separate components (left eye, right eye, left eyebrow, right eyebrow, mouth and nose).

2. Details of the network implementation

This paper designs a network model as shown in the figure 1.2, in which the first level uses the DCNNSF[7] network to predict the overall face characteristic sit point, and the other layer uses the new DCNN network, which is improved on the network infrastructure provided by the[2] [6] .The network takes the original image as input and performs regression policy on the coordinates of the desired feature points.

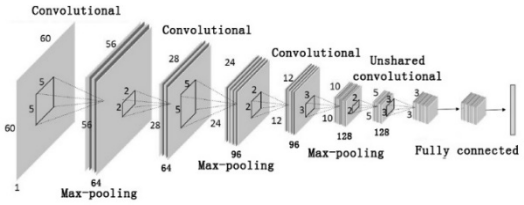

Fig. 1 DCNNSF-CFC Network Architecture

The input layer is a three stacked convolutional layer. Each convolutional layer applies multiple filters to the multi-channel input image and outputs response information. Let the input of the t convolutional layer as $\mathrm{I}t$, then output C can be get according to formula (1).

$$C_{i,j,k}^{t} = \left|tanh \sum_{x=0}^{h_t-1} \sum_{y=0}^{w_t-1} \sum_{z=0}^{c_t-1} I_{i-x,j-y,z}^{t-1} \cdot F_{x,y,k,z}^{t} + B_k\right| \quad (1)$$

Where I represents the input of the convolutional layer, F and B is the adjustable parameter. To make the system nonlinear, hyperbolic tangent and absolute value functions are used in filter response. After convolution, use Max-pooling as shown in formula (2):

$$I_{i,j,k}^{t} = \max_{0 \le x < d, 0 \le y < d} \left(C_{i \cdot d+x, j \cdot d+y, k}^{t}\right) \quad (2)$$

In order to achieve good robustness and greatly compensate for the information loss during the pool operation, this paper selects the max pooling layer, The convolutional layer is followed by the non-shared weight convolutional layer, that is, the Locally-Connected Layer. The final prediction will be generated by one or two fully connected layers and the parameters are adjusted to minimize the loss function as shown in the formula (3)

$$C_{i,j,k} = \left|tanh\left(\sum_{x=0}^{h-1} \sum_{y=0}^{w-1} \sum_{z=0}^{c-1} I_{i-x,j-y,z} \cdot F_{i,j,x,y,k,z} + B_{i,j,k}\right)\right| \quad (3)$$

3. DCNNSF-CFC Architecture

According to Figure 1.2, the hierarchical structure of the cascaded convolutional neural network is subdivided into the following 9 layers, and the specific work of each layer is shown as bellows:

3.1 Initial Prediction Layer

This layer leverages the geometric constraints of the global arrangement of facial components, and predict the initial positions of 68 facial feature points based on the facial information provided by the input image I, and finally output

$L^{Initial} = \left\{x_1^{Initial}, y_1^{Initial}, x_2^{Initial}, y_2^{Initial}, \ldots, x_{68,}^{Initial}, y_{68}^{Initial}\right\}$ which present 68 feature point coordinates.

This layer uses the DCNNSF network proposed by [7]. but [7] is used to predict the 5-point feature point face, this paper fine-tunes the DCNNSF network and the output of the final fully connected layer is 68-dimensional vector $L^{Initial}$

3.2 Bounding Estimation Box Layer

This layer divides the face into two parts, the contour point and the inner feature point of the face, and determines I according to the face information provided by the image.

Input: Image I .

Output: $\{(x_0^i, y_0^i), (x_1^i, y_1^i)\}$

and $\{(x_0^c, y_0^c), (x_1^c, y_1^c)\}$ which represent the upper-left corner of the two initial bounding boxes and the lower-right coordinates.

3.3 Contour-box Layer

This layer accurately determines the CB (Contour Boundary box) that contains 17 profile feature points.

Input: The initial boundary box $\{(x_0^c, y_0^c), (x_1^c, y_1^c)\}$ for the image I and face profile.

Output: Border box CB for the contour of face:.

When you crop an image within a bounding box, due to the instability of face detection, part of the feature point information may not fall within the bounding box, so the bounding box is enlarged by 15%. The CNN parameter configuration is shown in Table 3.1.

Table 1. Face Contour Boundary Box Prediction Layer Parameter Configuration

Layer name	Type	Enter dimensions	Nuclear size	Output dimension
Data	Input			60×60×3
Conv1	Convolution	60×60×64	5×5/1	56×56×64
Pool1	Max pooling	56×56×64	2×2/2	28×28×64
Conv2	Convolution	28×28×64	5×5/1	24×24×96
Pool2	Max pooling	24×24×96	2×2/2	12×12×96
Conv3	Convolution	12×12×96	3×3/1	10×10×128
Pool3	Max pooling	10×10×128	2×2/2	5×5×128
Unshared4	Local	5×5×128	3×3/1	3×3×160
Conv5s1	Fully connection	3×3×160		1×1×160
Conv5s2	Fully connection			1×1×256
Conv6	Fully connection			1×1×4

3.4 Contour-Points Layer

This layer will determine 17 contour feature points based on the contour boundary box of the face through CNN training, averaging them with the initialized 17 contour point coordinates detected by the first layer, and reaching to the final 17 contour points.

Input: Image I and face contour boundary box CB :$\{(x_0^{cb}, y_0^{cb}), (x_1^{cb}, y_1^{cb})\}$.

Output:$L^c = \{x_1^c, y_1^c, x_2^c, y_2^c, \dots, x_{17,}^c, y_{17}^c\}$

Where $(x_1^c, y_1^c), (x_2^c, y_2^c), \dots, (x_{17,}^c, y_{17}^c)$ stands for 17 face contour coordinates.

The network structure used by this layer is basically same as layer CB layer, the last full connection layer, i.e. L^c .

Table 2. Face Contour Point Detection Layer Parameter Configuration

Layer name	Type	Enter dimensions	Nuclear size/step length	Output dimension
Data	Input			60×60×3
Conv1	Convolution	60×60×64	5×5/1	56×56×64
Pool1	Max pooling	56×56×64	2×2/2	28×28×64
Conv2	Convolution	28×28×64	5×5/1	24×24×96
Pool2	Max pooling	24×24×96	2×2/2	12×12×96
Conv3	Convolution	12×12×96	3×3/1	10×10×128
Pool3	Max pooling	10×10×128	2×2/2	5×5×128
Unshared4	Local	5×5×128	3×3/1	3×3×160
Conv5s1	Fully connection	3×3×160		1×1×160
Conv5s2	Fully connection	1×1×160		1×1×256
Conv6	Fully connection	1×1×256		1×1×34

3.5 Face Inner Feature Point Boundary Box Prediction (Inner-Box Layer)

This layer will determine the inner boundary box IB which mark 51 feature points.

Input: The initial bounding box $\{(x_0^i, y_0^i), (x_1^i, y_1^i)\}$ for the image I and face profile.

Output: inner box IB for the outer contour of face :$\{(x_0^{ib}, y_0^{ib}), (x_1^{ib}, y_1^{ib})\}$.

Same as layer 3.3, the bounding box is enlarged by 15%. The network structure and the face profile boundary is basically the same as layer 3.3, so the parameter configuration is same as table 3.1.

3.6 Inner-Points Layer

In this layer, we will determine 51 contour feature points by using CNN training, then averages the initial 51 inner feature point coordinates detected by the initialization of the human face feature points, and get the final new 51 inner face feature.

Input: Image I and face inner feature point boundary box IB :$\{(x_0^{ib}, y_0^{ib}), (x_1^{ib}, y_1^{ib})\}$.

Output:$L^i = \{x_1^i, y_1^i, x_2^i, y_2^i, \dots, x_{51,}^i, y_{51}^i\}$..

Where $(x_1^i, y_1^i), (x_2^i, y_2^i), \dots, (x_{51}^i, y_{51}^i)$ is 51 inner feature point coordinates of a face.

The network structure used in this layer is essentially the same as the face profile feature point prediction layer Network structure, with parameters shown in Table3.3, and output only to 102-dimension vectors at the last full connection layer, i.e., L^i.

Table 3. Feature Point Detection Layer Parameter Configuration inside Face

Layer name	Type	Enter dimensions	Nuclear size/step length	Output dimension
Data	Input			60×60×3
Conv1	Convolution	60×60×64	5×5/1	56×56×64
Pool1	Max pooling	56×56×64	2×2/2	28×28×64
Conv2	Convolution	28×28×64	5×5/1	24×24×96
Pool2	Max pooling	24×24×96	2×2/2	12×12×96
Conv3	Convolution	12×12×96	3×3/1	10×10×128
Pool3	Max pooling	10×10×128	2×2/2	5×5×128
Unshared4	Local	5×5×128	3×3/1	3×3x160
Conv5s1	Fully connection	3×3×160		1×1×160
Conv5s2	Fully connection	1×1×160		1×1×256
Conv6	Fully connection	1×1×256		1×1×102

Then, we average the relative position coordinate of 51facial inner feature point obtained here with $L^{Initial}$ which obtained in initial detection layer, and get the new inner feature points of the face $L^{il} = \{x_1^{il}, y_1^{il}, x_2^{il}, y_2^{il}, \dots, x_{51}^{il}, y_{51}^{il}\}$.

3.7 Component Bounding Box Estimation Layer Estimation Layer

This layer will crop out 6 inner component boundary boxes (including left and right eye, left and right eyebrow, mouth, and nose) based on Lil.

Input: Image I and face inner feature points $L^{il} = \{x_1^{il}, y_1^{il}, x_2^{il}, y_2^{il}, \dots, x_{51}^{il}, y_{51}^{il}\}$

Output:6 inner component boundary boxes: Left eyebrow $\{(x_0^{ebl}, y_0^{ebl}), (x_1^{ebl}, y_1^{ebl})\}$ and right eyebrow $\{(x_0^{ebr}, y_0^{ebr}), (x_1^{ebr}, y_1^{ebr})\}$; Left eye $\{(x_0^{el}, y_0^{el}), (x_1^{el}, y_1^{el})\}$ and right eye $\{(x_0^{er}, y_0^{er}), (x_1^{er}, y_1^{er})\}$;Mouth $\{(x_0^{m}, y_0^{m}), (x_1^{m}, y_1^{m})\}$ and nose $\{(x_0^{n}, y_0^{n}), (x_1^{n}, y_1^{n})\}$.

Similarly, in order to retain more face information, when cropping the image within the bounding box, the program enlarges the bounding box by 15%. Taking the predicted internal feature points of each component as the center point, crop 6 bounding boxes. The upper left and lower right coordinates of the bounding box are the minimum and maximum values of the feature point coordinates contained in the corresponding internal components

$$(x_{min}\ , y_{min}\ ), (x_{max}\ , y_{max}\ )$$

3.8 Inner Component Feature Point Prediction (Component-Points Layer)

According to the bounding box of the internal components of the face, 6 feature points of the internal components of the face are predicted through CNN training. The output of each component is shown as output Li

Input: Image I and 6 faces inside the component boundary box.

Output:$L^i = \{x_1^i, y_1^i, x_2^i, y_2^i, \dots, x_{51,}^i, y_{51}^i\}$..

The output of each component is:

$L^{ebl} = \{x_1^{ebl}, y_1^{ebl}, x_2^{ebl}, y_2^{ebl}, \dots, x_5^{ebl}, y_5^{ebl}\}$,

$L^{ebr} = \{x_1^{ebr}, y_1^{ebr}, x_2^{ebr}, y_2^{ebr}, \dots, x_5^{ebr}, y_5^{ebr}\}$,

$L^{el} = \{x_1^{el}, y_1^{el}, x_2^{el}, y_2^{el}, \dots, x_6^{el}, y_6^{el}\}$,

$L^{er} = \{x_1^{er}, y_1^{er}, x_2^{er}, y_2^{er}, \dots, x_6^{er}, y_6^{er}\}$,

$L^n = \{x_1^n, y_1^n, x_2^n, y_2^n, \dots, x_9^n, y_9^n\}$,

$L^m = \{x_1^m, y_1^m, x_2^m, y_2^m, \dots, x_{20}^m, y_{20}^m\}$.

Where $(x_k^{ebl}, y_k^{ebl}), (x_k^{ebr}, y_k^{ebr}), (x_k^{el}, y_k^{el}), (x_k^{er}, y_k^{er}), (x_k^m, y_k^m), (x_k^n, y_k^n)$

Feature point coordinates corresponding to each internal component (6 for left and right eyes, 5 for left and right eyebrows, 20 for mouth and 9 for nose).

The network structure used in this layer is basically the same as the network structure of the facial contour feature point prediction layer. The difference is that the final fully connected layer outputs 12, 10, 40, and 18 one-dimensional vectors respectively.

Once all the facial components correspond to the feature point coordinates $L^{ebl}, L^{ebr}, L^{el}, L^{er}, L^m, L^n$ here, group them together to get the final inner feature point of the face:

$$L^{Inner} = (L^{ebl}, L^{ebr}, L^{el}, L^{er}, L^m, L^n)$$

3.9 Face Feature Point Output Layer (Output Layer)

Based on the results of the 3.4 face profile point detection layer output and the final result of the inner component feature point prediction layer output in 3.8, the final face 68 feature points are combined.

This layer enters the $L^{CL} = \{x_1^{CL}, y_1^{CL}, x_2^{CL}, y_2^{CL}, \dots, x_{17,}^{CL}, y_{17}^{CL}\}$ outer contour point of the human face and the feature point $L^{Inner} = (L^{ebl}, L^{ebr}, L^{el}, L^{er}, L^m, L^n)$ inside the face.

This layer output is the final face 68 feature points:

$$L^{ALL} = (L^{CL}, L^{Inner})$$

An excellent style manual for science writers is [7].

4. Experiments

4.1 Experimental environment and parameter settings

The experimental environment and the parameter settings used in this article are shown in Table 4. The predicted sample size of the experiment was 170,000 and the test sample was 45000.

Table 4. Experimental parameter settings

Parameters	Describe	Value
test_interval	Train again The number of iterations	2000
test_iter	Test the number of iterations once	500
batch_szie1	Single Processing Training Samples	100
batch_size2	Single processing Number of test samples	100
base_lr	Basic learning rate	0.01
lr_policy	The law of change of learning rate	step
gamma	Learning Rate Change Index	0.1
stepsize	Frequency of learning rate changes	100000

display	Screen display interval	100
max_iter	Maximum number of iterations	480000
Momentum	Momentum	0.9
weight_decay	Weight Decay	0.0005
snapshot	Save model intervals	5000
solver_mode	Mode selection	Gpu

4.2 Experimental Results and Analysis

In order to further prove the effectiveness and robustness of DCNNSF-CFC in the prediction of 68 feature points on the face, the experiment selected eight representative algorithms for comparison: DRMF-CLM [8], OPM-CLM [9] FF-AAM [10], BorMan-Regression [11], Cascade CNN [2], GM Methods [12], and RCPR Regression [1]

Since some network algorithms only disclose the model and not the source code, fair comparisons cannot be made. In addition, different methods of labeling use different numbers of facial feature point coordinates. Therefore, the experiment compares each model with the best results in the prediction of the respective number of facial feature points. The results are shown in Table 4.2. Among them, "Any" means that the author has published the code, so it is possible to train different numbers of facial feature points.

Table 5. Average error on three different data sets

Algorithm	Number of feature points	AFW	LFPW	Helen
DRMF	66	9.367	7.220	8.288
Opm	66	11.145	10.312	11.589
FF-AAM	Any	12.242	7.391	8.936
Cascaded CNN	5	5.446	5.765	3.913
GM-99	68	12.345	14.109	13.489
GM-1050	68	11.825	1333	13.418
BorMan	29	12.817	10.746	11.200
RCPR	Any	8.738	6.435	5.465
DCNNSF-CFC	68	6.993	5.360	5.839

It can be seen that the DCNNSF-CFC model has achieved better performance on all three databases. Compared with the traditional network, the main reason is that the DCNNSF-CFC model network has a strong deep learning ability, especially when there are many trainings. Samples can be feature learning, and then classified or detected. The cascaded CNN model that also uses the convolutional neural network has better performance in the image than the DCNNSF-CFC model, because CNN detects 5 feature points of the face: the center of the two pupils, the tip of the nose, and the two opposite Easy to detect corners of the mouth. The DCNNSF-CFC detects 68 feature points of the face. Under the same circumstances, the first-level network DCNNSF proposed in Chapter 3 of this paper has proved that it has the effect of not losing to the cascaded CNN in the prediction of five-point feature points of the face.

4.3 Experimental Presentation and Analysis

Figure 2, Figure 3, and Figure 4 respectively show the detection examples of DCNNSF-CFC algorithm on AFW data set, LFPW data set and Helen data set. In the experiment, the feature points are completely visible, the feature points are hidden by hair, glasses, etc. Points are hidden due to the camera angle of view, and feature points are hidden due to image cropping to detect the model.

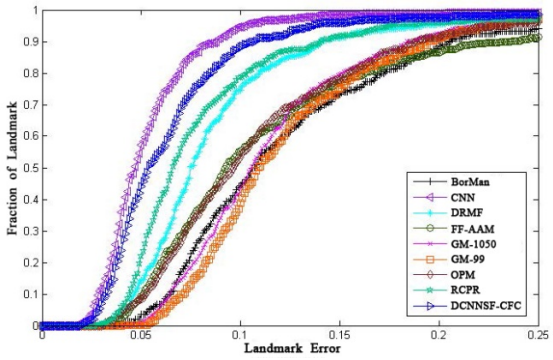

Fig. 2 Landmark error rate on AFW dataset

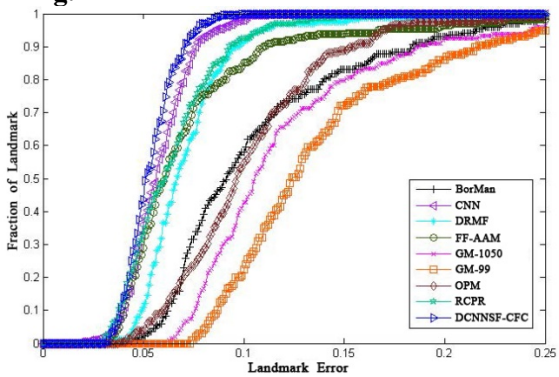

Fig. 3 Landmark error rate on LFPW dataset

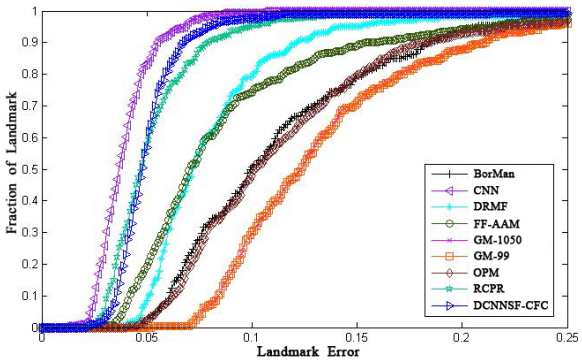

Fig 4. Landmark error rateon Helen dataset

It can be seen from the above examples that the network model proposed in this paper performs facial feature point location on the AFW data set, LFPW data set and Helen data set, and the feature point location performance on different face data sets has changed. In most cases, the facial feature points predicted by the model have a high accuracy rate. However, when some pictures are under extreme illumination and severe occlusion, the facial feature points that are partially occluded or in shadow are deviated from the prediction. In general, comprehensively comparing the face positioning performance of the algorithm, it can be seen that the DCNNSF-CFC network model is better and more robust than most traditional classic models.

5. Conclusion

This paper first describes the related concepts of cascaded regression and deep convolutional cascaded neural networks. Aiming at the problem that a single network model is usually not suitable for multi-feature point prediction, a global to local cascaded convolution is proposed on the basis of the DCNNSF model. The neural network DCNNSF-CFC adopts the "divide and conquer" strategy for 68 feature points, which divides them into prediction problems of contour points and internal

feature points of the face, and further classifies and predicts the internal feature points of the face to make the model to achieve higher accuracy.

Acknowledgments

Funding Information This Research Was Supported By University Distinguished Young Research Talents Training Program Of Fujian Province (Year 2018), Applied Practical Innovation Teaching Team Of Xiamen Institute Of Technology (Year 2019), Online And Offline Hybrid Course "Database Principle And Application" Of Xiamen Institute Of Technology (Year 2021).

References

[1] X. P. Burgos-Artizzu, P. Perona, P. Doll ar, Robust face landmark estimation under occlusion, in: Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE, 2013, pp. 1513-1520.

[2] Yang M, Kriegman D, Ahuja N (2002) Detecting faces in images: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(1):34-58.

[3] X. Xiong, F. De la Torre, Global supervised descent method, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2664-2673.

[4] Z. Zhang, P. Luo, C. C. Loy, X. Tang, Facial landmark detection by deep multi-task learning, in: Computer Vision-ECCV 2014, Springer, 2014, pp. 94-108.

[5] H. Lai, S. Xiao, Z. Cui, Y. Pan, C. Xu, S. Yan, Deep cascaded regression for face alignment, preprint arXiv:1510.09083, 2015.

[6] E. Zhou, H. Fan, Z. Cao, Y. Jiang, Q. Yin, Extensive facial landmark localization with coarse-to-fine convolutional network cascade, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013, pp. 386-391.

[7] Zheng Yinhuan, Wang Huan, Wang Jiaxuan, Chen Lingyu, Hong Qingqi. Deep convolutional neural network applied to the study of facial feature point detection.

[8] A. Asthana, S. Zafeiriou, S. Cheng, M. Pantic, Robust discriminative response map fitting with constrained local models, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013, pp. 3444-3451.

[9] X. Cao, Y. Wei, F. Wen, J. Sun, Face alignment by explicit shape regression, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 2887-2894.

[10] J. M. Saragih, S. Lucey, J. F. Cohn, Deformable model fitting by regularized landmark mean-shift, International Journal of Computer Vision 91 (2) (2011) 200-215.

[11] M. Valstar, B. Martinez, X. Binefa, M. Pantic, Facial point detection using boosted regression and graph models, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 2729-2736.

[12] X. Zhu, D. Ramanan, Face detection, pose estimation, and landmark localization in the wild, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 2879-2886.