Gray-Edge-HOG feature based cascaded learning for facial landmark detection

Wenhui Zhang*, Wentong Wang, Shuang Zhao, and Bin Sun

Beijing University of Technology, 100124 Beijing, China

Abstract. Compared with the traditional statistical models, such as the active shape model and the active appearance model, the facial feature point localization method based on deep learning has improved in accuracy and speed, but there still exist some problems. First, when the traditional deep neural network model targets a data set containing different face poses, it only performs the preprocessing through the initialized face alignment, and does not consider the regularity of the distribution of the feature points corresponding to the face pose during feature extraction. Secondly, the traditional deep neural network model does not take into account the feature space differences caused by the different position distribution of the external contour points and internal organ points (such as eyes, nose and mouth), resulting in inconsistent detection accuracy and difficulty of different feature points. In order to solve the above problems, this paper proposes a convolutional neural network (CNN) based on gray-edge-HOG (GEH) fusion feature.

1 Introduction

Facial feature point localization is a hot research topic in the field of computer vision, which play a key role in face recognition and face representation [1]. One of the great advantages of deep networks lies in the learning of abstract features. Sun et al.[2] proposed a facial feature point localization model based on three-level Deep Convolutional Network Cascade. The network took the images of different areas of the face (left eye, right eye, nose tip, and mouth corners) as input, cascaded multiple convolutional neural networks, and realized refined facial feature point localization by training the parameters of each level of the network. Zhang et al. [3] proposed the Coarse-to-Fine Auto-Encoder Networks. This method cascaded multiple auto-encoding networks, and characterized the local features of the facial shape from the global features of the facial appearance. This method benefited from the powerful non-linear expression ability of the deep auto-encoding network, achieved a better calibration effect than the ordinary cascaded regression model SDM [4] and DRMF [5] in the 300-W facial feature point localization challenge, and also improved the calibration speed.

According to the prior knowledge extracted from the existing image features, the Histogram-of-Oriented-Gradient (HOG) features highlight the characteristics of the image region, and the edge information can better highlight the contour features. Before extracting

* Corresponding author: zhangwh@emails.bjut.edu.cn

the abstract features from the face image, the edge and area information that is beneficial to facial feature point localization are first extracted according to the detection requirements, so as to make up for the lack of the original image for the detection task. Based on the above points of view, this paper proposes a convolutional neural network based on Gray-Edge-HOG (GEH) fusion to locate the facial feature points.

2 Gray-Edge-HOG based convolutional neural network

In this paper, a convolutional neural network based on GEH (Gray-Edge-HOG) feature fusion is proposed. This network can extract the GEH features that can enhance the edge effect and regional effect on face images, and the features of the semantic abstraction ability, and can realize the accurate detection of face feature points, especially the external contour points, through the training of the network model. The structure of the network is shown in figure 1.

Fig. 1. Structure of Gray-Edge-HOG (GEH) feature based convolutional neural network.

2.1 Data preprocessing

The preprocessing of training data includes two phases: resizing and pose estimation. For the resizing phase, we first extract the bounding box of facial landmarks of an image using the maximum and minimum coordinates of the ground truths of all landmarks, and then expand the bounding box by setting expanding ratios as 0.15,0.15,0.1,0.25 for four corners of the box, respectively. Finally, we normalize the images on $224 \times 224 \times 3$ resolutions.

In the online testing process, the preprocessing of testing data includes two phases: initial detection and resizing. For the initial detection phase, we apply the binary classification on each image patch of the sliding window, with the classifier given by the pre-trained support vector machine and the feature of each patch is given by the Histogram-of-gradient feature. The detection result is given by the patch with the highest score of the classifier. After the initial detection phase, we apply similar steps of training data for the resizing testing images.

2.2 GEH image generation

As Zhou et al.[6] point out, when a large number of facial landmarks is required, a single-model method tends to fail in capturing the underlying structure of the relative position of all landmarks. Typically, the contour points (referred to as 17 points on the contour) suffer from more noise and environmental fluctuations produced by background than the inner points (referred to as 51 points for eyes, eyebrows, mouth and nose). To tackle this problem, many researchers have argued that interdependent components are more capable of exploring visual features than color components of intensity images [7].

For the pedestrian detection analysis, Ouyang et al.[7] first pre-processed the image and constructed three new input channels instead of the original RGB image. This method enhanced the ability of the convolutional neural network to extract the edge features by considering the edge information of the original RGB image, and ultimately achieved pedestrian detection. Inspired by this method, this paper combines various features that reflect the edges of the face and the attributes of the region: the grayscale, edge, and gradient histogram features are first extracted, and then the above three features are mapped to three coordinate systems of the RGB color space, respectively. Finally, a GEH (Gray-Edge-HOG) fusion feature is formed. The following describes the above three features, respectively.

2.2.1 Gray component

Let $\mathbf{I} = \{\mathbf{I}_R, \mathbf{I}_G, \mathbf{I}_B\}$ be an intensity image, where $\mathbf{I}_R, \mathbf{I}_G, \mathbf{I}_B$ denote the red, green, blue channels of the image, respectively. To preserve the chromaticity and brightness of an image while reducing computation, we convert a color image to a gray image by using

$$\mathbf{I}_{\text{gray}} = 0.299\mathbf{I}_R + 0.587\mathbf{I}_G + 0.114\mathbf{I}_B \quad (1)$$

2.2.2 Edge component.

To preserve the edge feature of an image, we apply four-directional five-order templates for generating an edge image by first applying edge detection along four directions of the gray image, and then computing the pixel-wise magnitude of four images, and finally applying the min-max normalization. The detected edge images along four directions are given by:

$$\mathbf{I}_0 = \mathbf{T}_0 * \mathbf{I}_{\text{gray}}, \quad \mathbf{I}_{45} = \mathbf{T}_{45} * \mathbf{I}_{\text{gray}}, \quad \mathbf{I}_{90} = \mathbf{T}_{90} * \mathbf{I}_{\text{gray}}, \quad \mathbf{I}_{135} = \mathbf{T}_{135} * \mathbf{I}_{\text{gray}} \quad (2)$$

where $*$ denotes the convolution operator, $\mathbf{T}_0, \mathbf{T}_{45}, \mathbf{T}_{90}, \mathbf{T}_{135}$ denote the convolutional Kernel with respect to four directions, respectively, which are given by

$$\mathbf{T}_0 = \begin{pmatrix} 2 & 3 & 0 & -3 & -2 \\ 3 & 4 & 0 & -4 & -3 \\ 6 & 6 & 0 & -6 & -6 \\ 3 & 4 & 0 & -4 & -3 \\ 2 & 3 & 0 & -3 & -2 \end{pmatrix}, \quad \mathbf{T}_{45} = \begin{pmatrix} 0 & -2 & -3 & -2 & -6 \\ 2 & 0 & -4 & -6 & -2 \\ 3 & 4 & 0 & -4 & -3 \\ 2 & 6 & 4 & 0 & -2 \\ 6 & 2 & 3 & 2 & 0 \end{pmatrix}, \quad \mathbf{T}_{90} = \mathbf{T}_0^\top, \quad \mathbf{T}_{135} = \text{fliplr}(\mathbf{T}_{45}) \quad (3)$$

where $\text{fliplr}(\cdot)$ returns a matrix with columns of the input matrix flipped in the left-right direction. The edge image of $\mathbf{I}$ is then given by:

$$\begin{aligned} \mathbf{I}_{\text{edge}}(x, y) &= \sqrt{\mathbf{I}_0^2(x, y) + \mathbf{I}_{45}^2(x, y) + \mathbf{I}_{90}^2(x, y) + \mathbf{I}_{135}^2(x, y)} \\ \mathbf{I}_{\text{edge}}(x, y) &\leftarrow \frac{\mathbf{I}_{\text{edge}}(x, y) - \min \mathbf{I}_{\text{edge}}}{\max \mathbf{I}_{\text{edge}} - \min \mathbf{I}_{\text{edge}}} \end{aligned} \quad (4)$$

where (x, y) denotes the index of pixels of $\mathbf{I}$.

2.2.3 HOG component

We also apply Histogram-of-Oriented-Gradient (HOG) descriptor [8], which has wide applications in target detection, for an input image. To generate a two-dimensional matrix $\mathbf{I}_{\text{HOG}}$ describing the HOG feature of $\mathbf{I}$, we modify the original HOG algorithm [8] by assigning each pixel the angle of the direction which admits the greatest change among its neighborhood. The algorithm is given as follows: (a) Compute the gradient of each pixel $\nabla \mathbf{I}(x, y)$; (b) For each pixel (x, y) , we assign it with the orientation of the gradient of the pixel with the greatest magnitude among all pixels of a $\delta \times \delta$ neighborhood of (x, y) ; (c) Generate the matrix $\mathbf{I}_{\text{HOG}}$ by applying the min-max normalization for all the orientation angles of $\mathbf{I}$. The HOG matrix is computed via

$$\begin{aligned} (i', j') &= \arg \max_{(i, j) \in \mathcal{N}_{\delta}(x, y)} \|\nabla \mathbf{I}(i, j)\|_2 \\ \mathbf{I}_{\text{HOG}}(x, y) &= \tan^{-1}(\nabla \mathbf{I}(i', j')) \\ \mathbf{I}_{\text{HOG}}(x, y) &\leftarrow \frac{\mathbf{I}_{\text{HOG}}(x, y) - \min \mathbf{I}_{\text{HOG}}}{\max \mathbf{I}_{\text{HOG}} - \min \mathbf{I}_{\text{HOG}}} \end{aligned} \quad (5)$$

where $\mathcal{N}_{\delta}(x, y)$ denotes the δ -neighborhood of (x, y) of $\mathbf{I}$. Recalling the process of HOG feature, we see that the second identity of Eq. (5) is a special case of infinite orientation bins of the original HOG.

2.2.4 Feature extraction of convolutional layer

In this paper, the convolutional neural network is used to further extract the convolutional feature of the semantic abstraction from the GEH fusion feature. In order to preserve the image feature of the original image converted into the GEH image mode, the size of the face sub-image of the original image is taken as the input size, and the input of the convolution operation of the first layer is the size of 224×224 .

The convolutional kernel size is 7×7 , 4×4 , 3×3 , respectively, and the step size of the convolutional operation is set to 2. The j th feature graph x_j is calculated as: $x_j^l = f(b_j + \sum_i w_{ij} * x_i^{l-1})$ where l represents the layer number of the current network, w_{ij} represents the convolutional kernel parameter, b_j represents the bias parameter, w_{ij} and b_j are obtained by random normal initialization at the beginning of the experiment, f represents the activation function. In this paper, the activation function is Rectified Linear Unit (ReLU) function, the formula is as follow: $f(x) = \max(0, x)$.

The reason for selecting ReLU is that the ReLU function has a obvious advantage: the function has a linear characteristic, and the convergence of the gradient value obtained by using this function is faster than other activation functions. The pooling layer operation adopts the largest pooling method. For the characteristics of the GEH image input size, the pooling ranges of the first and third layers are set to 3×3 , and the step size is 3. While ensuring the completeness of the extraction features, this paper effectively reduces the feature dimension and the complexity of network training. The pooling range of the second layer is 2×2 , and the step size is 2. The activation function of the fully connectional layer is still the ReLU function. In order to improve the feature expression ability of the network and prevent overfitting of the network, this paper adopts the Dropout training strategy [9]. At two fully connectional layers, the dropouts p are all set to 0.5 making the probability of each output node of the network is $1 - p$ during the training phase, and effectively prevents overfitting in network training.

3 Experiments

3.1 Datasets

To verify the validity of our proposed method, this section will conduct experiments on the LFPW and Helen databases provided by the 300-W Challenge. The above two databases are face images obtained in a non-laboratory environment. The databases include 3365 face images, and each image is marked with 68 feature points. The databases include a variety of images, such as gestures, expressions, lighting, race, age, etc. In this paper, the training data includes 811 images from the trainset of the LFPW database and 2000 images from the trainset of the Helen database, a total of 2811 facial images. The test data includes 224 images from the testset of the LFPW database and 330 images from the testset of the Helen database, a total of 554 face images.

3.2 Experimental Results

Our proposed method mainly uses the Caffe framework to construct and train the network framework, and the Dlib open source toolkit [10] and OpenCV for face detection. Our method and the traditional deep convolutional neural network [6] adopt same network structure as introduced in Section 2.2.4.

We compare the feature extraction ability of GEH feature fusion based convolutional neural network (GEH-CNN) and traditional deep convolutional neural network (DCNN). Figure 2 shows the feature visualization of the output layers of the two neural networks. From the figure, we can see that the feature extracted by the GEH-CNN has more obvious edge contour information than the DCNN. The evaluation index of the point-to-point error is measured by the average estimation error index proposed by Yi Sun et al.[2].

Fig. 2. Feature visualization extracted by DCNN (top) and GEH-CNN (bottom).

Fig. 3. The error rate between the test position of facial landmarks and the real position based on LFPW and Helen.

We use two cases to compare and analyze the data, and verify the localization accuracy of the algorithm: (a) figure 3 shows the point-to-point errors between the test results and ground truth; (b) figure 4 shows the average error of the feature points and ground truth in different regions (left eyebrow, right eyebrow, left eye, right eye, nose, mouth, and outer contour). Figure 3(I) shows the test results on the LFPW dataset. The overall error of GEH-CNN is between 2% and 13%, most of which is between 4% and 9%. The overall error of the DCNN algorithm is between 3% and 17%, most of which is between 5% and 13%. Therefore, our proposed GEH-CNN is better than DCNN on the LFPW dataset. Figure 3 (II) shows the test results on the Helen dataset. The overall error of GEH-CNN is between 4% and 21%, most of which is between 5% and 15%. The overall error of DCNN is between 5% and 26%, most of which is between 6% and 20%. Therefore, our proposed GEH-CNN is better than DCNN in each region. In particular, the detection of external contour points has a lower average error.

The overall detection error rate of GEH-CNN is within 12%, and the average image detection error is 8.7%. Our proposed GEH-CNN can achieve better detection effect on face images under the influence of complex natural lighting, posture deflection, expressions and other factors. Therefore, our proposed method has good robustness and accuracy. It is worth noting that our proposed method achieves a certain improvement in the detection effect of the external contour points, which are more accurate.

Fig. 4. The mean error of each facial landmarks tested on LFPW and Helen

4 Conclusion

Our proposed method can effectively locate the feature points of the face. The convolutional neural network can better detect the outer contour points of the face with the obvious external contour representation of the GEH feature, and it based on GEH fusion feature greatly simplifies the network structure of feature points in face regions, and reduces the detection complexity. However, the face image is affected by lighting and other factors, some images have the problem that the outline feature of the human face is not obvious. In the future, we will further analyze the influence of lighting and shadows, and extract clear edge information.

References

1. K. Fukunaga, L. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Inf. Theory*, **21**(1), 32-40 (1975)
2. Y. Sun, X. Wang, X. Tang, Deep convolutional network cascade for facial point detection. *Computer Vision and Pattern Recognition*, 3476-3483 (2013)
3. J. Zhang, S. Shan, M. Kan, X. Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. *European Conference on Computer Vision*. 1-16 (2014)
4. X. Xiong, F. De la Torre, Supervised descent method and its applications to face alignment. *Computer Vision and Pattern Recognition*, 532-539 (2013)
5. A. Asthana, S. Zafeiriou, S. Cheng, M. Pantic, Robust discriminative response map fitting with constrained local models. *Computer Vision and Pattern Recognition*, 3444-3451 (2013)
6. E. Zhou, H. Fan, Z. Cao, Y. Jiang, Q. Yin, Extensive facial landmark localization with coarse-to-fine convolutional network cascade. *Computer Vision Workshops*, 386-391 (2013)
7. W. Ouyang, X. Wang, Joint deep learning for pedestrian detection. *Computer Vision*, 2056-2063 (2013)
8. G. Tsai, Histogram of oriented gradients. *University of Michigan*, **1**(1): 1-17 (2010)
9. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580* (2012)
10. Dlib C++ Library, <http://dlib.net>, (2018)