LI-7

Face detection based on gradient features and polynomial neural network

Linlin Huang Akinobu Shimizu Hidefumi Kobatake
Tokyo University of Agriculture & Techonology
{llhuang, simiz, kobatake}@cc.tuat.ac.jp

1. Introduction

Face detection is currently a very active research area because it has many potential applications, such as surveillance systems, intelligent human-computer interface, video conferencing etc.

2. System overview

To detect faces of variable sizes and locations, the detector needs to exa mine the input image in multiple scales. Each re-scaled image is scanned exhaustively to examine all windows of standard size (20x20 pixels). After the preprocessing procedure [1], the gradient features extracted from the local image of scanning window and gray scales form a vector, which is projected on the subspace learned by principal component analysis (PCA). The projection is used as the input of the polynomial neural network (PNN). The PNN is trained to give high output value for face pattern. Therefore, the test window with an output value higher than a threshold is considered as a face candidate. The overlapping detections within one scale or across different scales compete each other so that the test window with the highest output value is retained to be a detected face [1].

3. Gradient feature extraction and polynomial neural network

3.1 Gradient feature extraction

In frontal upright face image, eyes and mouth are mainly in horizontal direction while nose in vertical direction. Since direction features are more discriminatory, we extract gradient features from local image and generate five feature vectors to be the input of the classifier.

The Sobel operator is used to compute the gradient vector $\mathbf{u}(x,y)=[u_x,u_y]^T$.

The Sobel gradient features can be applied as input features. But in a face image, the facial features are not always exactly in horizontal or vertical directions. Therefore, four directions shown in Fig.1 are concerned, which divide the range into 8 regions (4 pairs). If the gradient vector lies between the two directions, e.g., Ω^2 region, it is decomposed onto the two directions, e.g., direction 2 and 3, shown in Fig.2.

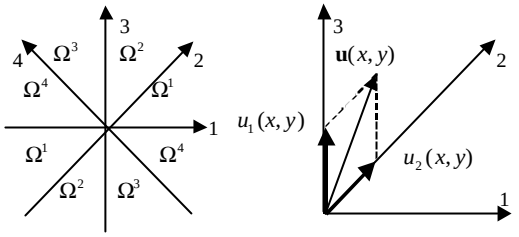

Fig. 1 Four directions

Fig.2 Decomposition

Consequently, four decomposed gradient features, say as $u_1(x,y)$, $u_2(x,y)$, $u_3(x,y)$, $u_4(x,y)$, are obtained from the two components of Sobel gradient vector $u_x(x,y), u_y(x,y)$.

The two components of Sobel gradient vector and the four decomposed gradient features are corresponding to six sub-images. The gray scale image and these six sub-images have the same size of 20x20 pixels and they are applied to generate five feature vectors:

(1)**gray:** After preprocessing, the gray levels of the gray scale image $f(x,y)$ are arranged to be a 368-dimensional feature vector[1].

(2)**sobel:** The two sub-images, $u_x(x,y), u_y(x,y)$, are normalized into 10x10 and masked to 92 pixels. The values of the pixels compose a 184-dimensional feature vector.

(3)**grad1**: The four sub-images, $u_1(x,y)$, $u_2(x,y)$, $u_3(x,y)$, $u_4(x,y)$ are normalized into 10x10 and masked to 92 pixels. The values of the pixels compose a 368-dim feature vector.

(4)**grad2**: Since in a face image, horizontal direction features are dominant, we exclude the first direction sub-image $u_1(x,y)$. The left three images $u_2(x,y)$, $u_3(x,y)$, $u_4(x,y)$ are normalized into 10x10 and masked into 92 pixels to compose a 276-diml feature vector.

(5)**gray-grad2:** The gray scale image $f(x,y)$ and the three decomposed gradient images, $u_2(x,y)$, $u_3(x,y)$ and $u_4(x,y)$ are normalized into 10x10 and masked to 92 pixels to compose a 368-dim feature vector.

3.2 Polynomial neural network

Denote the input pattern as $\mathbf{x}=(x_1,\cdots,x_d)^T$, the output of PNN [1] can be computed by

$$y(\mathbf{x}) = g\left(\sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d}\sum_{j=i}^{d} w_{ij} x_i x_j + w_0\right)$$

$g(\cdot)$ is a sigmoid function. The input vector is projected onto a feature subspace learned by PCA:

$$z_j = (\mathbf{x} - m)^T f_j \quad j = 1,2,\ldots m, \quad m < d .$$

m is the mean vector of face space. $f_j, j = 1,...,m$ are eigenvectors corresponding to the *m* largest eigenvalues. The distance from the feature subspace (DFFS) provides useful information for discrimination [1]:

$$D_f = \|\mathbf{x} - m\|^2 - \sum_{j=1}^{m} z_j^2 \cdot$$

Hence, we incorporate the DFFS into the PNN with hope to improve the detection performance:

$$y(\mathbf{x}) = g(\sum_{i=1}^{m} w_i z_i + \sum_{i=1}^{m}\sum_{j=i}^{m} w_{ij} z_i z_j + w^D D_f + w_0)$$

The connecting weights are updated by gradient descent to minimize the mean square error:

$$E = \frac{1}{2}\{\sum_{n=1}^{N_x}[y(\mathbf{x}^n) - t^n]^2 + l \sum_{w \in W - \{w_0\}} w^2\} = \frac{1}{2}\sum_{n=1}^{N_x} E^n$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) - h\frac{\partial E^n}{\partial \mathbf{w}}$$

where N_x is the total number of samples, t^n is the target output value, it takes value 1 for face sample and 0 for non-face sample. l is the coefficient of weight decay to restrict the size of connecting weights (excluding the bias). h is the learning rate, which is small enough and decreases progressively.

4. Experimental results

4.1 Training samples

29,900 face samples are generated from 2,990 real face images by changing the sizes, aspect ratio and reflection. The non-face samples are collected in three steps. In the first step, the non-face samples are collected by Euclidean distance between the test window and the mean vector of face space. In the second and the third steps, the non-face samples are collected by the PNN trained on face samples and non-face samples collected in the first step or collected in the first and second step.

4.2 Experimental results

In order to evaluate the performances of the five feature vectors as well as to compare with other system, the 23 cluttered images containing 149 faces used in MIT group [2] are tested. The results are given below and some detection examples are shown in Fig.3.

	True positives	False positives	Computation cost multiplications / window
MIT [2]	126	13	245,700
gray	126	22	41,850
Sobel	126	48	32,650
grad1	126	17	41,850
grad2	126	14	23,450
gray-grad2	128	11	41,850

5. Conclusion

The decomposed gradient features perform better than gray scales and Sobel gradient features. The results obtained by gray scales, Sobel gradient features, decomposed gradient features are not as good as that of [2]. However, the computation costs are much less than that of [2]. **gray-grad2** achieves better results than [2] while the computation cost is much less. This justifies that the combination of gray and decomposed gradient features is the most effective.

Reference

[1] L. L. Huang, A. Shimizu, Y. Hagihara, H. Kobatake, Face detection from cluttered images using a polynomial neural network, Proc. IEEE Conf. on Image Processing, 2001, pp. 667-669.

[2] K. K. Sung, T. Poggio, Example-based learning for view-based human face detection, IEEE Trans. Pattern Ana. & Mach. Intell, 20 (1) (1998) 39-50.

Fig.3 Detection examples