FACE DETECTION FROM CLUTTERED IMAGES USING A POLYNOMIAL NEURAL NETWORK

Lin-Lin Huang, Akinobu Shimizu, Yoshihiro Hagihara, Hidefumi Kobatake
Graduate School of Bio-Applications and Systems Engineering, Tokyo University of Agri. & Tech.
2-24-16 Naka-cho Koganei-shi, Tokyo 184-8588, Japan

ABSTRACT
In this paper, we propose a new method for face detection from cluttered images. We use a polynomial neural network (PNN) for separation of face and non-face patterns while the complexity of the PNN is reduced by principal component analysis (PCA). In face detection, the PNN is used to classify sliding windows in multiple scales and label the windows that contain a face. The PNN is shown to be powerful to discriminate between face and non-face patterns when trained with large number of samples. In experiments on images with simple or complex backgrounds, the proposed method has achieved high detection rate and low false positive rate.

1. INTRODUCTION
Face detection is an essential step for automatic face recognition systems and can also be used alone in security and interface applications. Despite its importance, this problem did not receive intensive interest until recently. Face detection is more difficult than it was expected due to the diverse variation of face appearance and the complexity of background images. In recent years, increasing efforts have been devoted to this problem and great progresses have been achieved. Nevertheless, it is still considered to be an unsolved problem.

The methods proposed for face detection so far can be coarsely grouped into four categories: template matching, geometrical models, statistical approach, and neural network approach. Frequently, template matching and geometrical models are combined in multistage knowledge-based systems [1-4], whereas the statistical approach [5-7] and the neural network approach [8-10] cannot be clearly separated because of their inherent connections. Generally, statistical and neural approaches achieve higher accuracy in detection because they are powerful of discrimination between face and non-face patterns when trained with large number of examples.

This paper proposes a new method for face detection, which uses a polynomial neural network for classification and PCA for dimensionality reduction. Due to the feature extraction capability of PCA (eigenface), the proposed method in effect integrates the advantages of eigenface method and neural network learning. The experimental results demonstrated that the framework of feature extraction followed by neural network classification was effective.

2. SYSTEM OVERVIEW
To detect faces of variable sizes and locations, the detector needs to examine the sliding windows of the test image in variable scales. Each sliding window is classified to be a face or a non-face. For convenience of processing, usually the input image is re-scaled into multiple images and in each re-scaled image, windows of standard size are processed. A standard window classified to be a face corresponds to a face box in the original image. This multi-scale strategy has been adopted in majority of previous works, e.g., [9,10]. Detected faces in different re-scaled images may correspond to overlapping face boxes in original image. In this case, they should be arbitrated so that the winner with higher similarity is retained. Also, in a re-scaled image, multiple detected face windows are arbitrated to retain the window of local maximum similarity.

In our work, the size of the standard window is 20x20 pixels. As in previous works, the corner pixels of a window
are masked (see Fig. 1) because they are highly unstable. The lighting condition of the masked window is ratified via subtracting the window image with an optimal fitting plane in sense of minimum square error. Then the gray scales are stretched by histogram equalization. The pixel intensities of the masked window compose a feature vector of 368 measurements for classification.

The classifier for face detection is a polynomial neural network (PNN) which has a single output unit to indicate whether an input pattern (masked window) is a face or not. PNN is very powerful in function approximation and regression yet the number of parameters is huge. To overcome this problem, the dimensionality of the input pattern is reduced by PCA. The PCA technique also plays the role of feature extraction so as to improve the efficiency of classification. As alternatives to PCA, the feature extraction can be accomplished by more efficient techniques [6,11] to further improve the detection performance.

Fig. 1. Masking of the normalized window

3. POLYNOMIAL NEURAL NETWORK

PNN can be viewed as a generalized linear classifier which uses not only the measurements of the input pattern but also the polynomials of the measurements as stimuli [12]. PNN has been successfully applied to handwritten character recognition [13], where PNN was shown to outperform MLP (multilayer perceptron) and RBF (radial basis function) network. The second-order PNN is also closely related to the Gaussian quadratic classifier since they both utilize the second-order statistics in pattern space [12,14].

Denote the input pattern as \( x = (x_1, x_2, \cdots, x_d)^T \), the output of PNN can be computed by

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

where \( g(\cdot) \) is a sigmoid activation function

\[
g(a) = \frac{1}{1 + \exp(-a)}.
\]

We can see that if the dimensionality of \( x \) is high, the number of connecting weights of PNN is too large. This can be overcome by dimensionality reduction via PCA. Denote the mean vector of face space as \( \mu \), the eigenvectors as \( \phi_j, j = 1, 2, \ldots, d \), corresponding to eigenvalues sorted in decreasing order, the features in the subspace of \( m (m<d) \) dimension can be obtained by

\[
z_j = (x - \mu)^T \phi_j \quad j = 1, 2, \ldots, m.
\]

The features are the projections of the input pattern onto the subspace spanned by \( m \) eigenvectors while the information in the complement space is omitted. PCA extracts the eigenvectors of feature subspace such that the approximation error is minimized. However, in object detection, the distance of the input pattern from the feature space provides useful information for discrimination:

\[
D_f = \|x - \mu\|^2 - \sum_{j=1}^{m} z_j^2.
\]

It is expected that the object patterns have small distance from feature space and non-object patterns have large distance. Finally, the PNN that we use has the output of

\[
y(x) = g \left( \sum_{i=1}^{m} w_i z_i + \sum_{i=1}^{m} \sum_{j>i}^{m} w_{ij} z_i z_j + w^D D_f + w_0 \right)
\]

where \( w \) and \( z \) represent the weight vector and the generalized input vector, respectively.

In training PNN on face and non-face examples, the connecting weights are updated by gradient descent to minimize the mean square error

\[
E = \frac{1}{2} \sum_{n=1}^{N} \left( y(x^n) - t^n \right)^2 + \lambda \sum_{w \in W - \{w_0\}} w^2 = \frac{1}{2} \sum_{n=1}^{N} E^n,
\]

where \( N_x \) is the total number of samples, \( t^n \) is the target output of sample \( x^n \), which takes value 1 for face examples and 0 for non-face examples. \( \lambda \) is the coefficient of weight decay to restrict the size of connecting weights (excluding the bias). By stochastic gradient descent, the weights are updated sequentially on each input pattern \( z^n = z(x^n) \) as

\[
w(n+1) = w(n) - \eta \frac{\partial E^n}{\partial w}.
\]

where \( \eta \) is the learning rate, which is small enough and
decreases progressively. The partial derivatives are computed by
\[ \frac{\partial E_n}{\partial w_k} = \begin{cases} [y(z) - r^*]y(1 - y)z_k + \lambda w_k, & w_k \in W - \{w_0\} \\ [y(z) - r^*]y(1 - y), & w_k \in \{w_0\}. \end{cases} \]

Since the PNN is a single layer network, the training process is very fast and the influence induced by the random initialization of weights is very less.

4. EXPERIMENTAL RESULTS

We used two image data sets in experiments to train the PNN classifier and test the detection performance. The first data set contains 3,487 images of single face with simple backgrounds. The second data set contains 130 images that have been used in CMU [10]. We used 2,989 images from the first set to extract face samples, 228 images from the first set and 14 scene images from the second set to extract non-face samples. In extracting samples from a face image, the hand cropped face box is scaled into standard window size and gives a sample vector. The image within the box is stretched and compressed to give two samples. The aspect ratio is changed to give two samples. The mirror images of the above five face samples with respect to the y axis are also included in the sample set. Hence one real face image generates 10 positive sample vectors. In total, the training data has 29,890 positive samples.

The negative (non-face) samples were collected in two stages. The Euclidean distance between the test window and the mean vector of positive samples was used to detect faces on the 228 single face images and 14 scene images. The window images that exceed a similarity threshold were sampled at uniform rate to obtain the first set of 75,885 negative sample. The PNN was trained on the positive and negative samples and the resulting classifier was used to detect faces from the 228 and 14 images and all window images exceeding the threshold 0.5 were added to the negative sample set. The PNN was then re-trained on the enhanced sample set and used to detect faces from test images.

The performances of PNNs with different dimension of feature subspace were tested on 270 single face images (the detector did not assume single face, however) and 111 CMU images (containing 491 entire faces in total) that are totally independent of the training images. In the CMU data set, besides the 14 scene images, another 5 images with extremely big or small faces were also excluded. The test images were examined at 10 scales and overlapping faces in same scale and different scales are arbitrated to retain the winning face only. The detection rates and false positive rates are given in Table 1. The false positive rate is the ratio of the number of false positives to the number of examined windows. We can see that as the complexity of PNN (dimension of feature subspace) increases, the detection rate does not vary significantly, but the false positive rate decreases evidently. Some test images with detected faces are shown in Fig. 2. From Fig. 2 we can see that the faces missed by the propose method are either unduly deformed or blurred in image quality, while the false positives resemble face patterns very much. In the literature, the group of Kanade at CMU had reported a detection rate of 86.6% with the false rate of 1/1051,888 [10]. Our result is not so good as theirs, but is still competitive to the results of other groups considering that we tested on a large number of complex images. Compared to the method of [10], our method is much simpler in implementation and the difference of detection performance only lies in very low quality images. For example, the method in [10] takes about 383 seconds to process a 320x240 pixel image (200MHz CPU) while in our method it only takes approximately 15 seconds (933MHz CPU).

Table 1. Face detection results

<table>
<thead>
<tr>
<th>m</th>
<th>Detection rate</th>
<th>False positive rate</th>
<th>Detection rate</th>
<th>False positive rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>99.26%</td>
<td>1/1,634,850</td>
<td>84.93%</td>
<td>1/135,177</td>
</tr>
<tr>
<td>60</td>
<td>99.26%</td>
<td>1/1,852,830</td>
<td>83.70%</td>
<td>1/228,942</td>
</tr>
<tr>
<td>80</td>
<td>99.26%</td>
<td>1/1,852,830</td>
<td>84.32%</td>
<td>1/246,722</td>
</tr>
<tr>
<td>100</td>
<td>99.26%</td>
<td>1/1,985,175</td>
<td>84.73%</td>
<td>1/275,048</td>
</tr>
</tbody>
</table>

5. CONCLUSION

The proposed method uses a neural network for classification and PCA for feature extraction. The experiments on simple and complex images demonstrated the effectiveness of the method. The performance can be further improved in two ways: supplementation of representative samples and incorporation of more efficient feature extraction techniques.
REFERENCES


Fig. 2 Examples of face detection