Simple and Fast Face Detection System Based on Edges

S. Anila*, N. Devarajan**

* Department of Electronics and Communication Engineering, Sri Ramakrishna Institute of Technology, Coimbatore, T.N., India. 
  e-mail: anilasatish@gmail.com

** Department of Electrical and Electronics Engineering, Government College of Technology, Coimbatore, T.N., India. 
  email: profdevarajan@yahoo.com

Submitted: 26/02/2010
Accepted: 23/03/2010
Appeared: 30/03/2010

Abstract—Face detection is an important first step to many advanced computer vision, biometrics recognition and multimedia applications, such as face tracking, face recognition, and video surveillance. In this paper, a faster face detection system is proposed with minimal features based on edges. Our proposed framework consists of three steps: initially the images are enhanced by applying median filter for noise removal and histogram equalization for contrast adjustment. In the second step the edge image is constructed from the enhanced image by applying sobel operator. Then a novel edge tracking algorithm is applied to extract the sub windows from the enhanced image based on edges. The rectangle features are calculated for the sub windows, and these feature values are fed into a trained Backpropagation Neural Network (BPN) to classify the sub-window as either face or non-face. The performance of the proposed method is compared with adaboost classifier. The result shows that our proposed method outperforms than the existing in terms of processing time in training and testing.

Keywords: face detection, edge detection, rectangle features, Backpropagation Neural Network

NOMENCLATURE

$E_{ij}$ - Edge image

$\mu$ - Mean

BPN – Backpropagation Network

$S$ – Sigmoidal function

$w_{ih}$ – Weight between input and hidden layer

$w_{ho}$ - Weight between hidden and output layer

$d$ – Error value

1. INTRODUCTION

Face detection is the method of discovering all possible faces at different locations with different sizes in a given image. It has numerous computer vision applications. Face detection involves many research challenges such as scale, rotation, and pose and illumination variation. The techniques used for face detection have been researched for years and much progress has been suggested in literature. Most of the face detection methods focus on detecting frontal faces with good lighting conditions. According to Yang, G. and Huang, T.S. (1997), these methods can be characterized into four types: knowledge-based, appearance-based, feature invariant, and template matching.

Knowledge-based methods use human-coded rules to model facial features, such as two symmetric eyes, a nose in the middle and a mouth underneath the nose Yang, G. and Huang, T.S. (1997). Feature invariant methods try to find facial features which are invariant to pose, lighting condition or rotation Sung, K.K. and Poggio, T. (1998). Skin colors, edges and shapes fall into this category. Template matching methods calculate the correlation between a test image and pre-selected facial templates Lanitis, A. Taylor, C.J. and Cootes, T.F. (1995). The last category, appearance-based, adopts machine learning techniques to extract discriminative features from a pre-labeled training set. The Eigenfaces method Turk, M. and Pentland, A. (2000) is the most fundamental method in this category. Recently proposed face detection algorithms such as support vector machines Kepenekci, B. and Akar, GB. (2004), neural networks Rowley, H.A. Baluja, S. and Kanade, T. (1998), statistical classifiers Schneiderman H. and Kanade, T. (2000) and AdaBoost-based face detection Viola P. and Jones, M J. (2001) also belong to this class. Appearance-based methods have accomplished noble results in terms of accuracy and speed. However, these methods are software-centric algorithms which do not take the hardware phase into
consideration. To make a robust real-time face detector, Viola and Jones proposed a method based on AdaBoost learning. They obtained both fast computation and high detection rates by introducing rectangle features, integral images, and cascade structures of classifiers. But it requires ineffective memory space to store integral images. Besides, high demand of random data access of memory is necessary Viola P. and Jones, M. J. (2004). Consequently, many improved methods were researched. Li S. Z. and Zhang, Z. Q. [2004] proposed FloatBoost method for training the classifier. Backtracking scheme was employed for removing unfavorable classifiers from the existing classifiers. Wu et al. carried out multiview face detection using nested structure and real AdaBoost, Wu, B. Ai, H. Huang, C. and Lao, S. (2004). Even though, the classifiers discussed above are faster and robust, still they are slow, since they are calculating the features for the entire image.

As seen in the literature, especially in appearance based methods, almost all the systems are classifying the faces starting with some base size. For example, in Adaboost classifier the base size is 24×24 Viola, P. and Jones, M. (2004). Also, the entire image is sub-sampled to identify the faces. In this case, for each sub-window more than 160,000 features have been calculated. In this paper, a novel edge tracking algorithm is proposed to eliminate the background and to reduce the number of sub-windows to be scanned while sub-sampling, thus the proposed method performs faster. Also there is no need to cascade the classifiers to eliminate non-faces. The proposed system has been implemented as three step process: initially the images are enhanced by applying median filter for noise removal and histogram equalization for contrast adjustment. In the second step the edge image is constructed by applying sobel operator. A novel edge tracking algorithm is used to extract the sub windows from the image to extract the rectangle features while skipping the backgrounds. The rectangle features proposed by Viola, P. and Jones, M. (2004) has been used for feature extraction. And the feature values are fed into a trained Backpropagation Neural Network (BPN) to classify the sub-window as either face or non-face.

This paper is organized as follows. Section 2 presents the preprocessing steps as enhancement, Section 3 presents the proposed edge detection and feature extraction methods. Section 4, presents the Backpropagation Algorithm to classify face and non-face patterns. Section 5 shows the experimental results of frontal face detection, and conclusions follows in Section 6.

2. PREPROCESSING

2.1 Enhancement

The face images may be of poor contrast because of the curbs of the lighting conditions. So histogram equalization is used to compensate for the lighting conditions and improve the contrast of the image. And images are sometimes corrupted by various sources of noise. The fine details of the image represent high frequencies, which mix up with those of noise.

When low-pass filtering is used, some details in the image may be erased as well. In this experiment, median filtering is used to suppress the noise. To apply Median Filtering, the value at a pixel is replaced by the median of the values in the neighborhood of the pixel. Given a set of n numbers \( \{x_1,...,x_n\} \) the ordered set is defined. Two-dimensional median filters can be defined for arbitrary sizes and shapes of filter windows \( W(i,j) \). The operation is defined as the intensity value of the centered pixel in which the window is replaced by its median value Chen, T. and Wu, HR. (2001, Xu , X. Miller, E.L. (2002).

2.2 Feature Extraction

The rectangle features proposed by Viola, P. and Jones, M. (2004) is used in this paper. There are three kinds of rectangle features. The value of a two-rectangle feature is the difference between the sums of the pixels within two rectangular regions. The regions have the same size and shape and are horizontally or vertically adjacent (Fig.1). A three-rectangle feature computes the sum within two outside rectangles subtracted from the sum in a center rectangle. Finally a four-rectangle feature computes the difference between diagonal pairs of rectangles. These rectangle features are computed using an intermediate representation for the image which we call the integral image as discussed in Viola, P. and Jones , M. (2004).

![Fig.1. Four types of rectangle features](image4)

3. PROPOSED EDGE TRACKING ALGORITHM

There are over 160,000 features in each sub-window in Viola Jones face detection method, Viola, P. and Jones , M. (2004). It is infeasible to compute the completed set of features, even if we can compute them very efficiently. According to the hypothesis that small number of features can be associated to form an effective classifier Viola, P. and Jones, M.J. (2001), the challenge now turns to be how to find these features.

![Fig. 2. (a) Face image , (b) Edge image](image5)
In the proposed work, initially the edges are detected from the image by using simple ‘sobel’ operator. The figure 2 shows the edge image. A novel edge tracking method is proposed to extract the face region while ignoring the background region. The algorithm is discussed below.

From the edge image, at each row, starting from the first pixel, our proposed tracking algorithm search for the first edge pixel. When it finds an edge pixel, that position is considered as top-left coordinate of the sub window. Then in the same row it searches for the next edge pixel that is considered as top-right coordinate. With these top-left and top-right coordinates, then it searches for the edge pixel towards column wise, where it finds an edge pixel at any one of the column is considered as bottom-left and bottom-right coordinates. With these four coordinates, the sub-window is extracted from the edge image and the mean value is calculated. If the mean value is zero, then the sub-window is considered as background and skipped to extract the next sub-window. If mean value is not equal to zero then with the same coordinates the corresponding sub-window is extracted from the original image and the rectangle features have been calculated.

The algorithm is given as follows:

**Edge Tracking Algorithm**

1. Load the edge image $E_{ij}$
2. For each row
3. For each column
   i. Search for top-left, top-right, bottom-left and bottom-right coordinates.
   ii. Extract the sub-window from the edge image.
   iii. Calculate its mean ($\mu$)
   iv. If $\mu=0$, then the sub-window is considered as background. Goto Step 3 to extract the next sub-window.
   v. Else, extract the corresponding sub-window from the original image and calculate the rectangle features.

The following figure illustrates our proposed edge tracking algorithm:

![Enhanced Image](a.png) ![Background Region](b.png)

Fig. 3. Outputs from Edge Tracking Algorithm

4. BACKPROPAGATION NETWORK CLASSIFIER

The classifier employed in this paper is a three-layer Backpropagation Neural Network, which contains 4 input neurons, 4 hidden neurons and one output neuron (4-4-1). The BPN optimizes the net for correct responses to the training input data set. More than one hidden layer may be beneficial for some applications, but one hidden layer is sufficient if enough hidden neurons are used.

![Three-Layer Backpropagation Network](c.png)

Fig. 4. A Three-Layer Backpropagation Network

For training the BPN, Initially the rectangle features are normalized between [0,1]. That is each value in the feature set is divided by the maximum value from the set. These normalized values are assigned to the input neurons. The number of hidden neurons is equal to the number of input neurons. And only one output neuron. Initial weights are assigned randomly between [-0.5 to 0.5]. The output from the each hidden neuron is calculated using the sigmoid function,

$$S_1 = \frac{1}{(1 + e^{-\lambda x})}$$

(1)

where $\lambda=1$, and $x = \sum w_{ih}k_i$, where $w_{ih}$ is the weight assigned between input and hidden layer, and $k_i$ is the input value. The output from the output layer is calculated using the sigmoid function,

$$S_2 = \frac{1}{(1 + e^{-\lambda x})}$$

(2)
where $\lambda = 1$, and $x = \sum w_{ho} S_i$, where $w_{ho}$ is the weight assigned between hidden and output layer, and $S_i$ is the output value from hidden neurons. $S_2$ is subtracted from the desired output. Using this error ($d$) value, the weight change is calculated as:

$$delta = d * S_2 * (1 - S_2)$$

(3)

And the weights assigned between input and hidden layer and hidden and output layer are updated as:

$$W_{ho} = W_{ho} + (n * delta * S_i)$$

(4)

$$W_{ih} = W_{ih} + (n * delta * k)$$

(5)

where $n$ is the learning rate, $k$ is the input values. Again calculate the output from hidden and output neurons. Then check the error ($d$) value, and update the weights. This procedure is repeated till the target output is equal to the desired output. The network is trained to produce a 0.9 output value for face regions and 0.1 output value for non-faces.

The non-face training set images are extracted with our edge tracking algorithm as a semi-automatic process. For every image, the edges are scanned and the sub-windows are shown to the user with two choices face or non-face. The rectangle features have been extracted for each sub-window and the class is stored based on the user selection. Totally from 1520 images, our proposed algorithm extract only 67054 sub-images that is 30291 face regions and 36763 non-face regions. The rectangle features extracted from these subimages are considered as input for the BPN classifier. The following table shows the detection rate and false positive rate of our proposed method compared with the standard AdaBoost learning method. The result shows that our method performs better than the standard without doing any changes in the adaboost algorithm.

The experiments are performed based on ten-fold cross validation method. The entire face image dataset is divided into ten folds of equal size. Each time, 9 folds have been selected for training and the rest is used for testing. The summary of the performance is given in the following table.

Table 1. Performance Analysis at each fold

<table>
<thead>
<tr>
<th>BioID Database</th>
<th>Proposed Edge based System Detection Rate (%)</th>
<th>False Positive Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>95.70</td>
<td>4.2</td>
</tr>
<tr>
<td>2</td>
<td>91.38</td>
<td>8.5</td>
</tr>
<tr>
<td>3</td>
<td>93.64</td>
<td>6.3</td>
</tr>
<tr>
<td>4</td>
<td>92.91</td>
<td>7.0</td>
</tr>
<tr>
<td>5</td>
<td>95.34</td>
<td>4.6</td>
</tr>
<tr>
<td>6</td>
<td>94.57</td>
<td>5.4</td>
</tr>
<tr>
<td>7</td>
<td>92.73</td>
<td>7.2</td>
</tr>
<tr>
<td>8</td>
<td>90.11</td>
<td>9.8</td>
</tr>
<tr>
<td>9</td>
<td>94.92</td>
<td>5.0</td>
</tr>
<tr>
<td>10</td>
<td>92.66</td>
<td>7.3</td>
</tr>
</tbody>
</table>

The overall performance is compared with Adaboost algorithm, in terms of detection rate, false positive rate and the number of sub-windows extracted to detect the faces. The results are shown in the following table. The main advantage of the proposed method is detecting the face with minimal number of scanning. Approximately the proposed method is 200 times faster than Adaboost.

Table 2. Comparison of Face Detection Performance

<table>
<thead>
<tr>
<th>BioID Database</th>
<th>Detection Rate (%)</th>
<th>False Positive Rate (%)</th>
<th>Maximum No. of Subimages Extracted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed Edge Based System with BPN</td>
<td>95.33 %</td>
<td>4.5 %</td>
<td>139</td>
</tr>
<tr>
<td>Adaboost</td>
<td>94.12 %</td>
<td>6.5 %</td>
<td>38400</td>
</tr>
</tbody>
</table>
6. CONCLUSION

The main contribution of this paper is to propose a method to construct a simple and fast face detection system. Initially the images are enhanced by contrast adjustment and noise removal. Then the images are divided into number of blocks to extract the rectangle features. The feature values are calculated or the block is considered based on a novel edge tracking algorithm. Thus the backgrounds are eliminated and the face regions can be localized faster. And the feature values are fed into a trained Backpropagation Neural Network (BPN) to classify the block as either face or non-face. From table II it is observed that the proposed technique performs better than the existing Adaboost.

REFERENCES


AUTHORS PROFILE

Ms. S. Anila, received her M.E Degree in Applied Electronics from Government College of Technology, Coimbatore in the year 2005. Currently she is pursuing her Ph.D under Anna University, Chennai. She is a life member of ISTE and IACSIT. Her research interests include Pattern Recognition and Face detection in Biometrics.

Dr. N. Devarajan, received the B.E (EEE) and M.E (Power Systems) degrees from GCT Coimbatore in the year 1982 and 1989. He received the Ph.D in the area of Control Systems in the year 2000. He has published 95 papers in national and international conferences. He has published 27 papers in international journals and 10 in national journals. Under his supervision currently 10 research scholars are working and 4 scholars have completed their Ph.D. His areas of interest are control systems, electrical machines and power systems. He is member of system society of India, ISTE and FIE.