Computer Vision Based Semi-automatic Algorithm for face detection

Ms. Srushti Girhe
Dept. of Elect. and Telecomm. Engg.
YCCE
Nagpur (India)
girhe.srushti@gmail.com

Mrs. Nita Nimbarte
Dept. of Elect. and Telecomm. Engg.
YCCE
Nagpur (India)
nitangp@gmail.com

Dr. A.S. Khobragade
Dept. of Elect. and Telecomm. Engg.
RGCER
Nagpur (India)
atish_khobragade@rediffmail.com

Abstract—Face detection is concerned with finding whether or not there are any faces in a given image and, if present, returns the image location and extent of each face. This is the first step of any face processing system. Although automatic face detection algorithms have improved over the years, we find user-selected faces more reliable, mainly because automatic face detection algorithms are not accurate in bounding box detection (location and size). However, human operator is preferred for this task .In this paper computer vision based semi-automatic algorithm for face detection is presented in which user interaction is limited to only marking the bounding boxes. Efficient Image Processing Algorithm stated in this paper is implemented using MATLAB and the results obtained are shown.

Keywords- *Bounding boxes, computer Vision, face detection.*

*****

I. Introduction

FACE detection is one class of object detection which has many applications in computer based vision such as scene surveillance, face recognition, face tracking, facial expression recognition, facial feature extraction, digital cosmetics, biometric systems, gender classification, clustering, attentive user interfaces, etc. The goal of face detection is to determine whether or not there are any faces in the image and, if present, return the image location and extent of each face. As digital images are going more and more widespread their intelligent or automatic examining is becoming exceptionally important. In this spirit, many face detection techniques have been researched for years. Most of the face detection methods focus on detecting frontal faces with good lighting conditions [2], [3], [4]. In order to detect frontal faces, bounding boxes are required around all faces in an image under investigation. To generate the bounding boxes, different automated algorithms are available [5]. However, for this task human operators are preferred because of following reasons-

a) It minimizes missing faces and false detections.
b) The scene context is very important in judging the lighting condition. Automated algorithm is less effective in this point of view.

For example, consider an image where all persons of interest are illuminated by flashlight. Then, the illuminants are expected to agree with one another. Contrarily, assume that a person in the foreground and a person in the background is illuminated by flashlight and ambient light respectively. Then, a difference in the color of the illuminants is expected. Such differences are hard to distinguish in a fully-automated manner, but can be easily detected and excluded by manual annotation. Thus, in this paper a computer vision based semi-automatic frontal face detection method is presented.

II. Problem definition AND objective

Aim is to detect the frontal face from an image under investigation and, if detected, extract each face from that image.

Given arbitrary image, may consist of either faces or non faces. Computer vision based Semi-automatic detector is designed that detects faces only and discards any non faces. Human interaction is limited to marking bounding boxes around faces in an image.

III. Proposed methodology

The remaining sections of the paper will discuss the implementation of the detector. Section 1 will detail the form of the features as well as a new scheme for computing them rapidly. Section 2 will discuss the method in which these features are combined to form a classifier. The machine learning method used, an application of adaboost, also acts as a feature selection mechanism. While the classifiers that are constructed in this way have good computational and classification performance, they are far too slow for a real-time classifier. Section 3 will describe a method for constructing a cascade of classifiers which together yield an extremely reliable and efficient face detector.

1. Haar Features

Our proposed algorithm uses haar features which are similar to convolution kernels, used to detect the presence of that feature in a given image. Each feature results in a single value which is calculated by subtracting the sum of pixels under white rectangle from the sum of pixels under black rectangle. Proposed algorithm uses 24 × 24 window as a base window size to start evaluating these features in any given image. If we consider all possible parameters of haar features like scale, position and type we end up calculating about 160000+ features in this window.

 ISSN: 2321-8169
130– 133

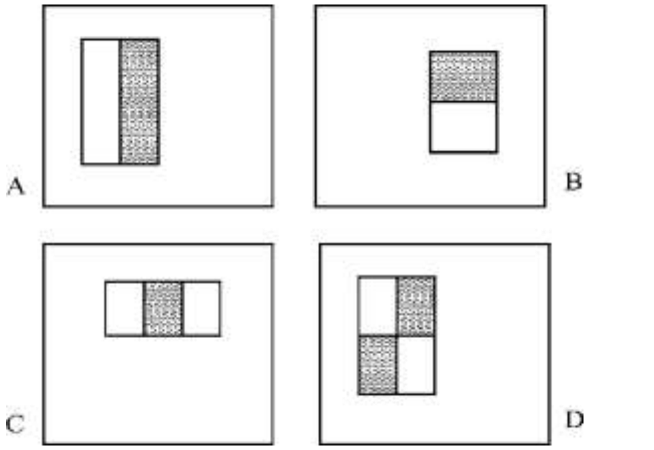

Fig.1 Two-rectangle features are shown in (A) and (B). Fig (C) shows a three-rectangle feature, and (D) a four-rectangle feature

2. Integral Image

It is an intermediate representation for the image by which rectangle features can be computed very rapidly. The integral image at location *x*, *y* contains the sum of the pixels above and to the left of *x*, *y*. Integral image allows for the calculation of sum of all pixels inside any given rectangle using only four values at the corners of rectangle.

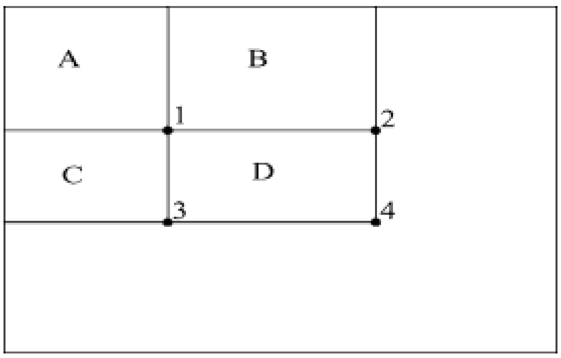

Fig.2 The sum of the pixels within rectangle D can be computed with four array references. The value of the integral image at location 1 is the sum of the pixels in rectangle A. The value at location 2 is A + B, at location 3 is A + C, and at location 4 is A + B + C + D. The sum within *D* can be computed as 4 + 1 − (2 + 3).

3. Adaboost Classifier

Adaboost is machine learning algorithm which helps in finding only the best features among all these 160000+ features. After these features are found, a weighted combination of all these features is used in evaluating and deciding any given window has a face or not. Each of the selected features are considered okay to be included if they can at least perform better than random guessing. These features are also called weak classifiers. Adaboost constructs a strong classifier as a linear combination of these weak classifiers.

Table1. The boosting algorithm for learning a query online. *T* hypotheses are constructed each using a single feature. The final hypothesis is a weighted linear combination of the *T* hypotheses where the weights are inversely proportional to the training errors

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0,1$ for negative and positive examples, respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0,1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots T$:

1: Normalize the weights,

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{k=1}^{n} w_{t,k}}$$

Weight error of weak classifier $\left(h_j\left(x, f_j, p, \theta\right)\right)$ is,

$$\varepsilon_j = \sum_{i=1}^{n} w_i \left| h_j(x_i) - y_i \right|.$$

2: Select the best weak classifier with respect to the weighted error $\varepsilon_t = \min_{f,p,\theta} \sum_{i=1}^{n} w_i \left| h_j(x_i) - y_i \right|$.

3: Define $h_i(x) = h_j(x, f_t, p_t, \theta_t)$ where f_t, p_t, θ_t are minimizers of ε_t .

4: Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i},$$

where $e_i = 0$ if x_i example is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{e_t}{1-e_t}$.

5: The final classifier is

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq 0.5 \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases}$$

where $\alpha_t = \log \frac{1}{\beta_t}$.

4. Cascading

The basic principle of face detection algorithm is to scan the detector many times through the same image-each time with a new size. Even if an image should contain one or more faces it is obvious that an excessive large amount of evaluated sub-windows would still be negatives (non faces). So, the algorithm should concentrate on discarding non-faces quickly and spend more time on probable face regions. Hence, a single strong classifier formed out of linear combination of all best features is not a good to evaluate on each window because of computational cost. Therefore, a cascade classifier is used which is composed of stages each containing a strong classifier. So, all the features are grouped into several stages where each stage has certain number of features. The job of each stage is used to determine whether a given sub-window is definitely not a face or may be a face. A given sub-window is immediately discarded as not a face, if it fails in any of the stages.

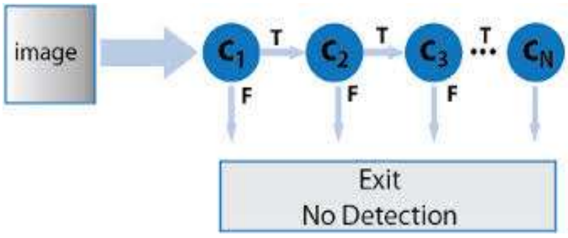

Fig. 3 Cascading

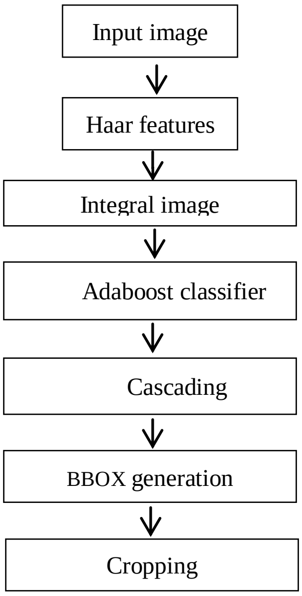

Fig. 4 Algorithm for face detection

IV. EXPERIMENTAL RESULTS

Above algorithm for face detection is executed using MATLAB and desired results are obtained.

Steps carried:

Step 1: The input image I, must be a grayscale or true color (RGB) image.

Step 2: Define and set up your cascade object detector using the constructor that detect objects using the Viola-Jones algorithm.

detector = vision.CascadeObjectDetector

Step 3: Call the step method with the input image, I, the cascade object detector object.

It will returns BBOX, an M-by-4 matrix defining M bounding boxes containing the detected objects.

Each row of the output matrix, BBOX, contains a four-element vector, [x y width height], that specifies in pixels, the upper-left corner and size of a bounding box.

Step 4: Annotate detected faces with shape, position and label. We can set the shape to 'rectangle' or 'circle', and insert it at the location specified by the matrix, Position. The input, label, can be a cell array of ASCII strings.

Step 5: Each detected face from an input image is then cropped.

IMAGE RESULTS:

Fig. 5 Detected faces

(a) (b) (c)

Fig. 6 (a), (b), (c) each face is extracted from a given arbitrary image

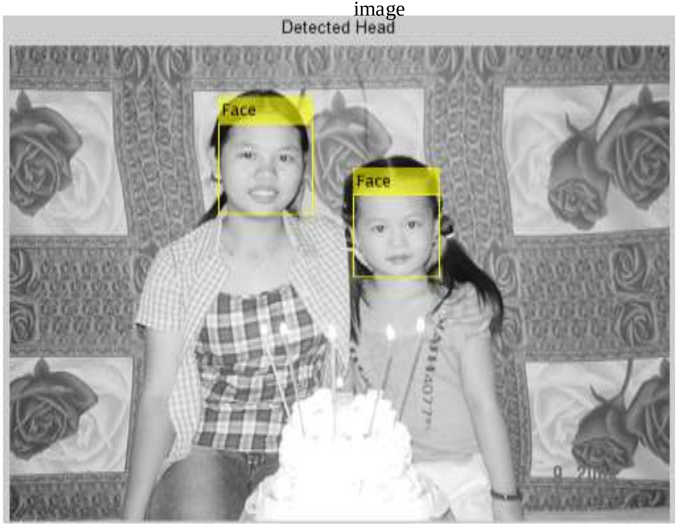

Fig. 6 Detected faces

Fig. 7 (a), (b) each face is extracted from a given arbitrary image

V. CONCLUSION AND FUTURE SCOPE

Specifically, the algorithm is able to robustly detect frontal faces in an image under investigation with high accuracy and efficiency. We are succeeded in obtaining desired results.

Main challenge for us in near future is to develop Computer Vision based algorithms for texture-based and gradient-based feature extraction for all face regions. In addition, we believe that our current face detection algorithm has a scope of improvement.

REFERENCES

[1] Viola, Paul and Michael J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. Volume: 1, pp.511–518

[2] H. Rowley, T. Kanade and S. Baluja. "Neural network-based face detection". In IEEE Patt. Anal. Mach. Intell., volume 20, pages 22–38, 1998

[3] D. Roth, M. Yang, and N. Ahuja. "A snowbased face detector". In Neural Information Processing 12, 2000.

[4] H. Schneiderman and T. Kanade. "A statistical method for 3D object detection applied to faces and cars". In International Conference on Computer Vision, 2000.

[5] W. R. Schwartz, , D. Harwood, A. Kembhavi and L. S. Davis, “Human detection using partial least squares analysis,” In Proc. IEEE Int. Conf. Computer Vision (ICCV), 2009, pp. 24–31.

[6] N. Allezard, J. Begard and P. Sayd. "Real-time human detection in urban scenes: Local descriptors and classifiers selection with adaboost like algorithms". In CVPR Workshops, 2008.

[7] Y.-T. Chen and C.-S. Chen. "Fast human detection using a novel boosted cascading structure with meta stages" Image Processing, IEEE Trans. on, 17(8):1452–1464, 2008.