International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

Face Detection for Authentication using Haar Classifiers

Alpana Deshpande†*, Pranjali Dashpute†, Sama Chaudhary† and S.B. Wankhade†

†Computer Engineering, Rajiv Gandhi Institute of Technology, Mumbai, India

Accepted 15 March 2016, Available online 23 March 2016, **Vol.6, No.2 (April 2016)**

Abstract

Face detection is the essential front end of any face recognition system, which locates and segregates face regions from cluttered images, either obtained from video or still image. It also has numerous applications in areas like surveillance and security control systems, content-based image retrieval, video conferencing and intelligent human-computer interfaces. It is a more reliable, fool proof alternative for image captcha or mail based authentication to prove that a human is using the system, hence making it less prone to security threats. Most of the current face recognition systems presume that faces are readily available for processing. However, in reality, we do not get images with just faces. We need a system, which will detect, locate and segregate faces in cluttered images.

Keywords: *Face detection, authentication, Viola Jones, HAAR classifiers*

1. Introduction

[1]We will be using feature based detection procedure. A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image. For example, let us say we have an image database with human faces. It is a common observation that among all faces the region of the eyes is darker than the region of the cheeks. Therefore a common haar feature for face detection is a set of two adjacent rectangles that lie above the eye and the cheek region. The position of these rectangles is defined relative to a detection window that acts like a bounding box to the target object (the face in this case).

Since it is a web based application, it should run effortlessly on a variety of systems without prior installations. For this purpose, we have used JavaScript to implement the face detection software and XML along with CSS and PHP to implement the website.

We have used face detection as a security measure. Because of this, it is essential that the program should accurately detect human faces and non-human faces for every iteration during runtime.

It is necessary for web applications to be speedy. In order to achieve this, the compilation and speed of runtime must be optimum.

The program must be fool proof. It should not detect human faces where there are not any as this would destroy the purpose of the project.

*Corresponding author: **Alpana Deshpande; S.B. Wankhade** is working as **HOD**

The application should efficiently perform the task that it is intended for. The compilations should be speedy as well as accurate and serve the purpose of authentication

2. Literature Review

Face detection is a wide and extensively studied subjects. Various algorithms have been implemented in order to detect human faces in real time. The automatic face detection algorithm (Yogesh Tayal *et al*, 2012) is applied on a wide variety of images taken under different lighting conditions and with different backgrounds. The images also have area containing skin from other parts of the body such as hands, necks and areas with color very similar to that of skin. These areas get classified as skin.For a 380×270 size of image, the total time taken by the algorithm was 2.30 seconds. The histogram is formed using a training set of over 4,90,000 pixels drawn from various sources on the internet. The training set contained skin pixels of people belonging to different races**.** Orthogonal wavelet templates (Chirag Patel *et al*, 2012) may also be used for implementation. Orthogonal wavelet considerer details of images and its multi-resolution representation. Template making procedure consider lots of face images. After that orthogonal wavelet transform is applied on that each image and make one template of face after averaging all wavelet transform images coefficients, and using matching algorithm that template is matched with test images and thus face is detected in images. Experimental results indicate that this approach is more efficient and accurate for robust domain. Two step face detection algorithms (Ridhi Jindal *et al*, 2013) involve application of skin filters and

detecting faces through skin, in this the binary skin map and the original image together are used to detect faces in the image. The technique relies on thresholding the skin regions properly so that holes in face regions will appear at the eyebrows, eyes, mouth, or nose.

This paper presents detection of faces by means of implementing the Haar algorithm. Haar-like features are evaluated through the use of a new image representation that generates a large set of features and uses the boosting algorithm Ada Boost. (Faiza Ahmad *et al*, 2012)To reduce degenerative tree of the boosted classifiers for robust and fast interferences only simple rectangular Haar-like features are used that provides a number of benefits. Implementation of a system that used such features would provide a feature set that was far too large, hence the feature set must be only restricted to a small number of critical features which is achieved by boosting algorithm.

3. HAAR algorithm (Viola Jones algorithm)

The face detection algorithm proposed by Viola and Jones is used as the basis of our design. The face detection algorithm looks for specific Haar features of a human face. When one of these features is found, the algorithm allows the face candidate to pass to the next stage of detection. A face candidate is a rectangular section of the original image called a sub-window. Generally this sub-window has a fixed size (typically 24×24 pixels). This sub window is often scaled in order to obtain a variety of different size faces. The algorithm scans the entire image with this window and denotes each respective section a face candidate. The algorithm uses an integral image in order to process Haar features of a face candidate in constant time. It uses a cascade of stages which is used to eliminate non-face candidates quickly. Each stage consists of many different Haar features. Each feature is classified by a Haar feature classifier. The Haar feature classifiers generate an output which can then be provided to the stage comparator. The stage comparator sums the outputs of the Haar feature classifiers and compares this value with a stage threshold to determine if the stage should be passed. If all stages are passed the face candidate is concluded to be a face. There are four main stages of detecting faces by using Haar algorithm (M Gopi Krishna *et al*, 2012).

- Haar Feature Selection
- Creating an Integral Image
- Classifier Training
- Cascading Classifiers

3.1 Haar Feature Selection

Features are nothing but common observations we can make regarding to the human facial structure. For instance the forehead area is brighter than the rest of the face. Haar features are rectangular structures for identification of these features. The value of a feature is the difference between the integral sum of the dark rectangular area and the integral sum of the white rectangular area.

Value = Σ (pixels in black area) - Σ (pixels in white area)

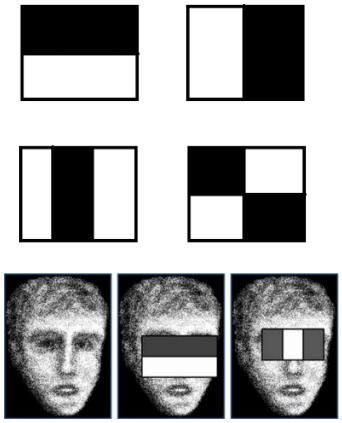

Fig. 1 Haar features

3.2 Integral image

To calculate the integral sum of a rectangular area, we first need to calculate the integral sum of each point on the image. The integral image at each point (x,y) is the sum of the pixels above and to the left of the image. This can be done easily in a single computation.

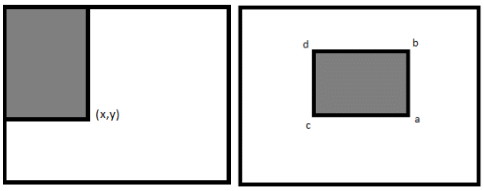

Fig. 2 Integral image

On calculating the integral image at each pixel (x,y) in the image, the integral value of rectangular haar features are calculated by taking into consideration the integral image of the four points- a ,b ,c and d of the rectangle.

$$\sum rect = a - b - c - d$$

Thus the integral image is calculated for the dark area and the white area of each of the Haar features.

3.3 Classifier Training

The Viola Jones algorithm uses the AdaBoost algorithm for selecting the best features and training the classifiers that use them. The weak classifiers are used linearly in order to form strong classifiers that give minimum false positives and 100% true positives.

$$h(x) = sign(\sum_{j=1}^{M} a_j h_j(x)$$

Set of N positive and negative training images with their labels (x^i, y^i) are taken as input. If image i is a face $y^i = 1$, if not $y^i = -1$. Assign a weight $w_t^i = \frac{1}{N}$ to each image I. For each feature f_j with j=1,....M

- Renormalize the weights such that they sum to one.
- Apply the feature to each image in the training set, then find the optimal threshold and polarity θ_j, s_j that minimizes the weighted classification error. That is

$$\theta_j, s_j = \arg\min \sum_{i=1}^{N} w_j^i \, c_j^i$$

- The weights for the next iteration are reduced for the images I that were correctly classified.
- Set the final classifier to

$$h(x) = sign(\sum_{j=1}^{M} a_j h_j(x))$$

3.4 Cascading Classifiers

Cascading the classifiers is what ensures speedy detection in the Viola-Jones algorithm. Cascades can be best described as funnelling. The top of the funnel uses weak classifiers to eliminate negatives and push the possible positives further down the funnel. As the funnel narrows, the classifiers get more and more complex while the number of false positives reduce gradually.

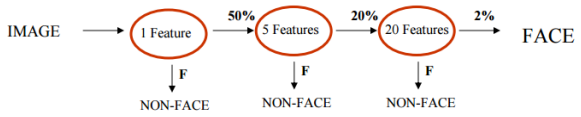

Fig. 3 Cascading Classifiers

Each level in the cascade omits images which are negatives (non faces). The images which may be positives are moved to the next level wherein the haar features get more complex. This reduces the time consumed as images which are definite negatives are omitted more quickly and are not subjected to complex haar features.

The final output of the cascade is an image with the face, with only a 2% chance of false positives.

Conclusion

By integrating face detection with a website, we develop an intricate system that would be an efficient way to check if a user is a human. This would also protect users of a website from malicious programs trying to obtain sensitive user data. Such a system would be of utmost importance in web applications such as online banking where leaking of sensitive data could cost companies and banks millions.

Because the application is for authentication, it is necessary that the program be accurate and without loopholes. For this purpose we will use HAAR classifiers as it accurately removes background images and separates faces from the background. The image would be taken from a webcam and hence, it would be impossible to find a way around the authentication procedure. HAAR classifiers also combine speed with accuracy and so, it does not reduce the overall speed of the web application. This is necessary for a good user experience.

Next, to ensure that the web application is available to every user, we need to provide platform independence. We do this by using JavaScript to implement face detection using HAAR classifiers. JavaScript will be integrated with the XML website and so, it will achieve platform independence.

References

Yogesh Tayal, Ruchita Lamba, Subhransu Padhee(2012), Automatic Face Detection using Color based Segmentation International Journal of Scientific and Research Publications, Volume 2, issue 6, June 2012

Chirag Patel, Sanjay Garg(2012) Robust Face Detection using Fusion of Haar and Daubechies Orthogonal Wavelet Template International Journal of Computer Applications, Volume 46, Issue 6, May 201

Ridhi Jindal, Anuj Gupta, Dr Sonia Vatta(2013), Face Detection using Digital Image Processing, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 11, November 201

Faiza Ahmad, Aaima Najam and Zeeshan Ahmed (2012), Image-based Face Detection and Recognition International Journal of Computer Science Issues Volume 9, Issue 6, No 1, November 20

M Gopi Krishna, A. Srinivasulu (2012), Face Detection System On AdaBoost Algorithm Using Haar Classifiers, International Journal of Modern Engineering Research(IJMER), Volume 2, Issue 5, Sept-Oct 2012