IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 3, June-July, 2013
ISSN: 2320 - 8791
www.ijreat.org

Face detection using boosted Cascade of features
Using Viola-Jones

[1]Amit Debnath, [2]Madhumita Debbarma,[3]Swapan Debbarma
[1]CSE Department,Tripura University (A Central University) Agartala, Tripura, India.
[2]MBA ,CMJ University, Shillong, Meghalaya, India
[3]CSE Department, National Institute of Technology, Agartala, Tripura, India.

Abstract—**During the recent years, high resolution cameras and video streams has had a significant on communication and in entertainment world. This paper describes a face detection frame work which is capable of processing of images extremely fast and achieve high detection rate. There are three key contributions in this paper. First is the a new modified image representation that is called "Integral image" it allows detector to compute very fast. The second one is the classifier which is built by modified AdaBoost, it uses very important features from a very large set of features. The third one is the combination of classifier that is called "cascade". The cascade provides a very effective mechanism that helps achieve high classification with low cpu utilization.**

Keywords- Boosting, cascade, face detection

I. INTRODUCTION

The purpose of this paper is to implement the algorithm presented by viola-jones and reconstruct the face detection algorithm. The ideal goal of any face detection algorithm is to perform well as per human inspecting. This algorithm capable of functioning in all constrain. This framework is motivated on face detection. we have constructed a frontal face detection system which achieves detection and false positive rates which are equivalent to best publish result (Schneiderman and kanade, 2000). This face detection is clearly different from previous approaches in its ability to detect faces. It works on 384 by 288 pixel images; faces are detected on 15 frame per second. In other detection system, to achieve high frame rate pixel color in color images is used. Our system use only grey scale image to achieve high frame rate.

There are three main key component of our face detection framework. We will discuss each of these three key components in sub-sequent section below.

The first key component of this paper is a new way to represent the images called *integral image* that allows us a fast feature evaluation. Inspired by the work of Papageorgiou et. Al. (1998) .our system doesn't work directly with image intensities like these authors instead we use set of feature to get high success rate.

The contribution of second key component is simple and effective classifier that is built by selecting small number of important feature from a huge library using AdaBoost (Freund and Schapire).

The third key component of this paper is combining successively more complex classifier in a cascade structure. It increases the speed of the detector.

A. *What is face detection?*

Face detection is a computer technology that determines the locations and sizes of human faces in arbitrary (digital) images. It detects facial features and ignores other things, such as buildings, trees and bodies. Face detection can be regarded as a specific case of object class detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belong to a given class. Face detection can be regarded as a more general case of face_localization. In face localization, the task is to find the locations and sizes of a known number of faces (usually one). In face detection, one does not have this additional information**.**

B. *Why face detection?*

Face detection is used in biometrics, often as a part of (or together with) a facial recognition system. It is also used in video surveillance, human computer interface and image database management. Some recent digital cameras use face detection for autofocus. Face detection is also useful for selecting regions of interest in photo slideshows that use a pan-and-scale Ken Burns's effect. A webcam can be integrated into a television and detect any face that walks by. The system then calculates the race, gender, and age range of the face. Once the information is collected, a series of advertisements can be played that is specific toward the detected race/gender/age.

Face detection is also being researched in the area of energy conservation.

Overview

In this paper we are going to discuss the three main component and remaining section will ducuss about the implementation of the decoder and the related theory, experiments regarding this. Section III we will discuss about features and new approach to compute them rapidly. In section IV we will discuss the method in which these features are combined to form a classifer. AdaBoost a meachine learning method act as a feature selection mechanism. Section V will describe a method for constracting a cascade of classifier which is relaible and efficient for face detection. section VI will describe implementation and results.

II. FEATURES

Our face detection metod based on the value of simple features. Its use feature rather than pixels. The most common reason is that features can act as a code ad-hoc

IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 3, June-July, 2013
ISSN: 2320 - 8791
www.ijreat.org

domain knowledge. It is difficult to learn a finite quantity of finite data. Another advantage of using features rather than pixel is feature based system can operate faster than pixel based system. As feature can easily extracts cmpare to pixel.

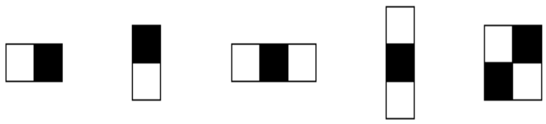
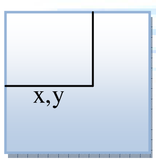


Fig.1 Different types of feature

In this paper we use three kinds of features. The value of *two rectangular* features is the difference between the sum of the pixels between two rectangular regions. A *three rectangle* feature computes the sum within two outside rectangle subtracted from the sum in a center rectangle and a *four rectangle* feature computes the difference between diagonal pair of rectangle.

III.A Integral Image

The integral image computes a value at each pixel (x,y) that is the sum of a pixel values above and to the left of (x,y), inclusive. This can be computed in one pass through the image.



Face detection algorithm is to turn the input image into an integral image. This is done by making each pixel equal to the entire sum of all pixels above and to the left of the concerned pixel This is demonstrated in Figure 2. A summed area table(also known as an integral image) is an algorithm for generating the sum of values in a rectangular subset of a grid.

1	1	1
1	1	1
1	1	1

Input image

1	2	3
2	4	6
3	6	9

Integral image

Fig.2 The integral image at location x,y contains the sum of the pixels above and to the left of x,y inclusive

$$ii(x,y) = \sum_{x' \le x, y' \le y} i(x',y'),$$

Where $ii(x,y)$ the integral is image and $i(x,y)$ is original image

$$s(x,y) = s(x,y-1) + i(x,y)$$

$$ii(x,y) = ii(x-1,y) + s(x,y)$$

(where $s(x,y)$ is the cumulative row sum, $s(x,-1)$ =0 and $ii(x,y) = ii(x-1,y) = 0$ the integral image can be compute in one pass over the original image.

The integral image is in fact the double integral of the image (first along rows and then along columns). The second derivative of the rectangle (first in row and then in column) yields four delta functions at the corners of the rectangle. Evaluation of the second dot product is accomplished with four array accesses.

The sum area table is an accumulation of pixel values, starting from the upper left and moving towards the lower right of an image. The value at any point (*x*, *y*) in the table is the sum of all the pixels above and to the left of that point.The summed area table can be computed efficiently in a single pass over the image, using the fact that the value in the summed area table at (*x*, *y*)

$$sum(x,y) = i(x,y) + sum(x-1,y) + sum(x,y-1) - sum(x-1,y-1)$$

The above equation is used to create the integral image from the original monochrome picture. The resulting integral image is then used to calculate Haar feature values using only 4 sums

$$\sum_{\substack{A(x)<x'<=O(x)\\ A(y)<y'<=O(y)}} i(x',y') = sum(A) + sum(C) - sum(B) - sum(D)$$

Using the integral images any rectangular sum can be computed in four array references.

III. Modified AdaBoost

AdaBoost is a machine learning Boosting algorithm capable of contracting a strong classifier with the help of week classifier.
It is a meta algorithm can be used in conjunction with many other learning algorithm to improve their performance. AdaBoost is adaptive in the sense that subsequent classifiers built are tweaked in favor of those instances misclassified

by previous classifiers. AdaBoost is sensitive to noisy data and outliers
A weak classifier is mathematically described as:

$$h(x,f,p,q) = \begin{cases} 1 & if \ pf(x) > pq \\ 0 & otherwise \end{cases}$$

Where x is a 24*24 pixel sub-window, f is the applied feature, p the polarity and the threshold that decides whether x should be classified as a positive (a face) or a negative (a non-face). Since only a small amount of the possible 160.000 feature values are expected to be potential weak classifiers the AdaBoost algorithm is modified to select only the best features.

- Given examples images $(x1, y1), \dots (x_m y_n)$ where $y1 = 0,1$ for negative and positive examples
- Initialize weights $w_{1,i} = \frac{1}{2_m}, \frac{1}{2_l}$ for $y1 = 0,1$, where m and l are the numbers of positive and negative examples
- For $t = 1, \dots, T$:

I. Normalize the weights,

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

II. Select the best weak classifier with respect to the weighted error:

$$\varepsilon_t = \min_{f,p,q} \sum_i w_i \mid h(x_i, f, p, q) - y_i \mid$$

III. Define $h_i(x) = h(x, f_t, p_t, q_t)$ where f_t, p_t and q_t are the minimizers of ε_t

IV. Update the weights :

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

Where $e_i = 0$ if examples x_i is classified correctly and $e_i = 1$ otherwise, and

$$\beta_t = \frac{\varepsilon_t}{1 - \varepsilon_t}$$

-The final strong classifier is:

$$C(x) = \begin{cases} 1 \text{ if } \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 \text{ otherwise} \end{cases}$$

Where $\alpha_t = \log \frac{1}{\beta_t}$

An important part of the modified AdaBoost algorithm is the determination of the best feature, polarity and threshold. There seems to be no smart solution to this problem and Viola-Jones suggest a simple brute force method. This means that the determination of each new weak classifier involves evaluating each feature on all the training examples in order to find the best performing feature.

V. Cascade Classifier

The basic principle of the Viola-Jones 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 the evaluated sub-windows would still be negatives (non-faces). This realization leads to a different formulation of the problem: Instead of finding faces, the algorithm should discard non-faces.

The thought behind this statement is that it is faster to discard a non-face than to find a face. With this in mind a detector consisting of only one (strong) classifier suddenly seems inefficient since the evaluation time is constant no matter the input. Hence the need for a cascaded classifier arises.

The cascaded classifier is composed of stages each containing a strong classifier. The job of each stage is to determine whether a given sub-window is definitely not a face or maybe a face. When a sub-window is classified to be a non-face by a given stage it is immediately discarded. Conversely a sub-window classified as a maybe-face is passed on to the next stage in the cascade.

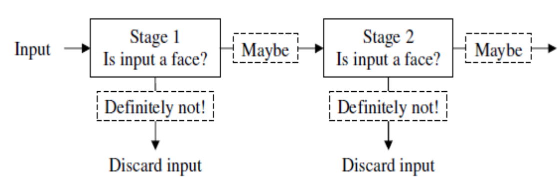


Fig.3 Cascade classifier

In a single stage classifier one would normally accept false negatives in order to reduce the false positive rate. However,for the first stages in the staged classifier false positives are not considered to be a problem since the succeeding stages are expected to sort them out. Therefore Viola-Jones prescribes the acceptance of many false positives in the initial stages. Consequently the amount of false negatives in the final staged classifier is expected to be very small.

Viola-Jones also refers to the cascaded classifier as an attentional cascade. This name implies that more attention (computing power) is directed towards the regions of the image suspected to contain faces.

It follows that when training a given stage, say *n*, the negative examples should of course be false negatives generated by stage *n*-1

VI. Cascade Creation

F0 = 1
i = 0
While Fi > Ftarget and i < nStages
i = i + 1
Train Classifier for stage i

Initialize Weights
Normalize Weights

Pick the (next) best weak classifier

Update Weights

Evaluate fi

if fi > f

go back to Normalize Weights

Combine weak classifiers to form the strong

Stage classifier

Evaluate

Weight for each

positive sample 0.5/m

negative sample 0.5/n

m – number of positive samples (200)

n – number of negative samples (100)

VII. Training of a Cascade classifier

The cascade training process involves two types of tradeoffs. In most cases classifiers with more features will achieve higher detection rates and lower false positive rates. But At the same time classifiers with more features require more time to compute. In principle one could define

i) the number of classifier stages

ii) The number of features in each stage, and

The false positive rate of cascade is:

$$F = \prod_{i=1}^{K} f_i$$

Where F is the false positive rate of the cascade classifier, K is the numbers, and f_i is the false positive rate of the i th classifier on the examples that get through to it. The detection rate

Where D is the detection rate of the cascade classifier, K number of classifier and d_i detection rate of the i th classifier on the examples that get through to it.

•User selects values for f , the maximum acceptable false

positive rate per layer and d, the minimum acceptable detection rate per layer.

• User selects target overall false positive rate, *F_target* .

• P = set of positive examples

• N = set of negative examples

• $F0$ = 1.0; $D0$ = 1.0

• $i = 0$

• while $Fi > F\_target$

– $i \leftarrow i + 1$

– $ni = 0$; $Fi = Fi{-}1$

– while $Fi > f \times Fi{-}1$

∗ $ni \leftarrow ni + 1$

∗ Use P and N to train a classifier with ni features using AdaBoost

∗ Evaluate current cascaded classifier on validation set to determine Fi and Di .

∗ Decrease threshold for the ith classifier until the current cascaded classifier has a detection rate of at least $d \times Di{-}1$ (this also affects Fi)

– $N \leftarrow \emptyset$

– If $Fi > F\_target$ then evaluate the current cascaded detector on the set of non-face images and put any false detections into the set N

VIII. Experiment result

In order to explore the feasibility of the cascade approach two simple detectors were trained: a monolithic 200-feature classifier and a cascade of ten 20-feature classifiers. The first stage classifier in the cascade was trained using 500 faces and 100 nonface sub-windows randomly chosen from non-face images. The second stage classifier was trained on the same 5000 faces plus 5000 false positives of the first classifier. This process continued so that subsequent stages were trained using the false positives of the previous stage.The monolithic 200-feature classifier was trained on the union of all examples used to train all the stages of the cascaded classifier. Note that without reference to the cascaded classifier, it might be difficult to select a set of non-face training examples to train the monolithic classifier. We could of course use all possible sub-windows from all of our non-face images, but this would make the training time impractically long. The sequential way in which the cascaded classifier is trained effectively reduces the non-face training set by throwing out easy examples and focusing on the "hard" ones. However, there is a big difference in terms of speed. The cascaded classifier is nearly 10 times faster since its first stage throws out most non-faces so that they are never evaluated by subsequent stage.

IX. Results

The face training set consisted of 4916 hand labeled faces scaled and aligned to a base resolution of 24 by 24 pixels. The faces were extracted from images downloaded during a random crawl of the WorldWideWeb. Some typical face examples are shown in The training faces are only roughly aligned. The training faces are only roughly aligned. This was done by having a person place a bounding box around each face just above the eyebrows and about half-way between the mouth and the chin. This bounding box was then enlarged by 50% and then cropped and scaled to 24 by 24 pixels

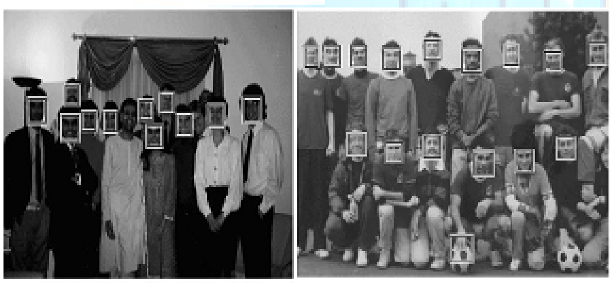


Fig.3 **Output of face detector on a number of test images from the MIT+CMU test set**

X. Conclusion

We have demonstrated detectors for in-plane rotated faces and for profile faces. Together these detectors handle most face poses encountered in real images. This work confirms that the Viola-Jones framework can handle non-frontal, non-upright faces despite some previous doubts along these lines. We have also presented a general method for selecting among a set of detectors while scanning an input image. This method works well and preserves the speed advantage of the Viola-Jones detectors.

REFERENCES

[1].PAUL VIOLA and MICHAEL J. JONES "*Robust Real-Time Face Detection"* International Journal of Computer Vision 57(2), 137–154, 2004 _c 2004 Kluwer Academic Publishers.

[2].Henry A. Rowley, Shumeet Baluja, and Takeo Kanade "Neural Network-Based Face Detection" (PAMI, January 1998).

[3].Diego A. Socolinsky, Joshua D. Neuheisel, Carey E. Priebe, Jason DeVinney, David Marchette "Fast Face Detection with a Boosted CCCD Classifier".

[4].Jan Sochman Jirı Matas "AdaBoost with Totally Corrective Updates for Fast Face Detection"

[5].Jones, M.; Viola, P. "Fast Multi-view Face Detection" MITSUBISHI ELECTRIC RESEARCH LABORATORIES.

[6]. 'Implementing the Viola-Jones face detection algorithm' Ole Helvig Jensen.

[7].Paul Viola and Michael Jones "*Robust Real-time Object Detection"* SECOND INTERNATIONAL WORKSHOP ON STATISTICAL AND COMPUTATIONAL THEORIES OF VISION – MODELING, LEARNING, COMPUTING, AND SAMPLING VANCOUVER, CANADA, JULY 13, 2001.

[8].Viola, P. and Jones, M "*Rapid Object Detection Using a Boosted*" MITSUBISHI ELECTRIC RESEARCH LABORATORIES TR2004-043 May 2004.

[9]. *"Viola Jones Simplified"* By Eric Gregori.

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

[11].K. Sung and T. Poggio. Example-based learning for view-based face detection. In *IEEE Patt. Anal. Mach. Intell.*, volume 20, pages 39–51, 1998.