IJCSC 0973-7391 Volume 7 • Number 1 Sept 2015 - March 2016 pp.76-82 available online at www.csjournalss.com

# Face Detection and Tracking: A Comparative study of two algorithms

Sonia Mittal[1], Chirag Shivnani[2]
[1]Assistant Professor, Nirma University, Ahmedabad, India
[2]MCA student, Nirma University, Ahmedabad, India
Sonia.mittal@nirmauni.ac.in, 14MCA165@nirmauni.ac.in

**Abstract:** This paper explains two object detection and tracking algorithms – Viola Jones and Camshift. . It comprises the overview of the algorithms, the features and stages involved, the advantages and dis-advantages, face detection using both algorithms and finally a comparison between detecting multiple faces using both algorithms.

## 1. Introduction

**1.1 Object-class Detection:** Object-class detection is a field of computer vision that deals with detection objects that belong to a particular class – humans, buildings, or cars – in digital videos of images. The detection of the object's class works on the fact that every object has its own special features that are unique to that particular object that helps in identifying the class of the object – for example, all circles are round. Object-class detection, using these special features, detect the object class.

### 1.2 Face Detection Overview

Face localization deals with the task of finding the locations and sizes of a known number of faces. Face detection can be regarding as a general case of face localization which in-turn can be regarded as a specific case of object-class detection. The concept behind face detection is that there is a database which contains a variety of face images and a face is processed and matched bitwise against these face images in the database and if a match is found that we can say that a face is detected.

A classifier is a method for determining the likely class of an unknown object based on the number of instances of each of the classes known as training set. In face detection, a classifier is used and trained using positive images i.e. images that contain a face and negative images, i.e. images that do not contain a face. In the training process, features are extracted from the training set which can be expressed as a vector of measurements and after the classifier is trained and the features extracted, the features are used to detect faces in a new set of digital images.

### 1.3 Face Detection Applications

Facial Recognition: It is used in biometrics and also in video surveillance and image database management.Photography: Modern digital cameras use face detection for auto-focus.Marketing: Marketers are taking interest in face detection. A webcam can be integrated into a television and detect any face that walks by. Information such as the race, gender and age range is calculated and advertisement shown geared toward that particular person.

### Edge Detection

The algorithm mentioned in this paper uses an underlying technique known as edge detection. It is a set of methods which are used to identify points in a digital image at which the image brightness changes sharply.

Consider an ideal case where the result of applying an edge detector to an image results in a set of connected curves that outlines the boundaries of an object. After applying the edge detection algorithm, the amount of data to be processed has been reduced significantly as we have to process the part of the image that lies within the boundaries only. All the information that isn't relevant to the object detection is filtered out.

Convolution is a function derived from two given functions by integration that expresses how the shape of one is modified by the other. Convolution matrix, or mask is a small matrix useful for blurring, sharpening, embossing, edge-detection. This is accomplished by means of convolution between a kernel and an image.

## 2. Face Detection and Tracking Algorithm

### 2.1 Viola Jones

#### 2.1.1 Introduction

The face detection algorithm, Viola-Jones [1] is the first object detection algorithm that provides competitive object detection rates in real-line and was proposed by Paul Viola and Michael Jones [1] in the year 2001. The algorithm is widely used to detect faces in digital images and videos but can also be used to detect a number of other object classes.

The motivation behind Viola-Jones algorithm was to develop an algorithm using which a computer can successfully detect a face efficiently and accurately. The algorithm limits itself to full vide frontal upright faces, i.e. in order for the face to be detected it must point towards the camera and it should not tilt to any side.

**Characteristics of the algorithm:**

**Robust:** The algorithm has a very high detection rate(true-positive rate) and very low false-positive rate[1].

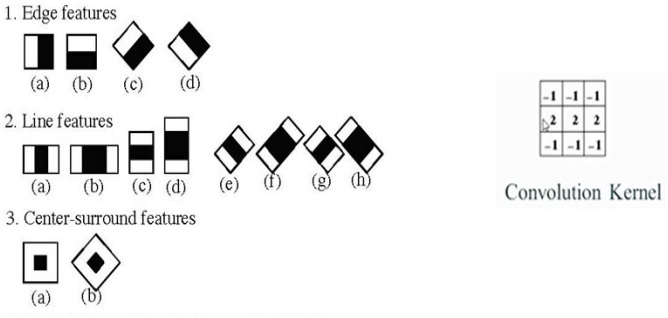
**Real Time:** At least 2 frames per second are processed thus making it a quick and an efficient algorithm.

The algorithm comprises of four stages:

- Haar Features Selection
- Creating Integral Image
- Adaboost Training Algorithm
- Cascade Classifiers

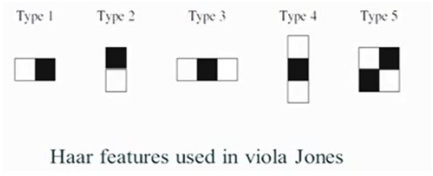
#### 2.1.2 Haar Features

Haar-like features are digital image features used in object detection[1].Haar features are similar to convolution kernels which are used to detect the presence of a feature in the given image.




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.

**Main Haar Features Used in Viola Jones:**




$$f(x,y) = \sum_i p_b(i) - \sum_i p_w(i)$$

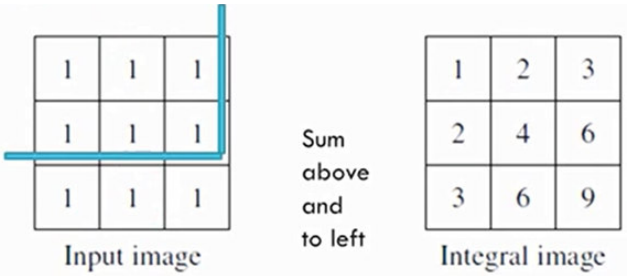
The black region in a harr feature is replaced by +1 and white region is replaced by -1.

Viola jones algorithm uses a 24x24 window as the base window size to start evaluating these features in any given image. (Example: Type 1)

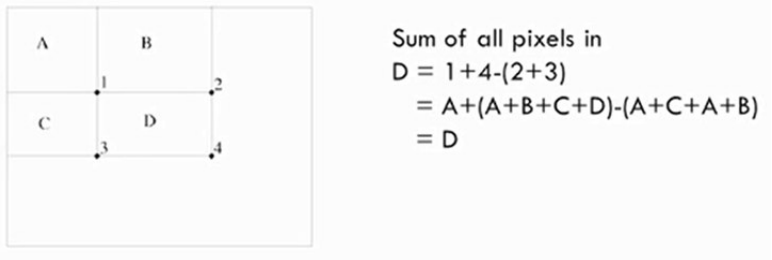
If we consider all possible parameters of Haar features like position, scale and type, we end up calculating about 160,000+features in this 24x24 window. (Problem: Inefficient to calculate 160k+ features in every window.)

### 2.1.3 Integral Image

It is time consuming to sum up all the black region pixels and white region pixels at every step.Viola-Jones algorithm tries to solve this problem by what is known as integral image[1].The algorithm introduces the concept of integral image to find the sum of all the pixels under a rectangle with just 4 corner values instead of summing up all the values.Generating integral image: in an integral image the value at pixel(x,y) is the sum of pixels above and to the left of (x,y).




Integral image allows for calculating of sum of all pixels inside any given rectangle using only four values at the corners of the rectangle.




### 2.1.4 Adaboost

Adaboost [6] is a machine learning algorithm which helps in finding only the best features among all the 160,000+ features.After these features are found, a weighted combination of all these features is used in evaluating and deciding if any given window (24x24) has a face or not. Each of the selected feature is considered to be included if they can at least perform better than random guessing.

These features are also called as weak classifiers[6]. Adaboost constructs a strong classifier as a linear combination of these weak classifiers.

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots$$

Strong classifier  Weak classifier

$$h(\mathbf{x}) = \operatorname{sign}\left(\sum_{j=1}^{M} \alpha_j h_j(\mathbf{x})\right)$$

In a standard 24x24 pixel sub-window, there are a 162,336total of possible features.A weak learner is defined to be a classifier which is only slightly correlated with the true classification (it can label examples better than random guessing). In contrast, a strong learner is a classifier that is arbitrarily well-correlated with the true classification.

Weak classifier is something that would at least perform better than random-guessing. If given 100 faces, it would at least be able to detect 50 faces.

Adaboost Algorithm[6]:

**Input:** Set of $N$ positive and negative training images with their labels $(\mathbf{x}^i, y^i)$. If image $i$ is a face $y^i = 1$, if not $y^i = -1$.

1. Initialization: assign a weight $w_1^i = \frac{1}{N}$ to each image $i$.
2. For each feature $f_j$ with $j = 1, \ldots, M$
   - 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 $\theta_j, s_j$ that minimizes the weighted classification error.
   - Assign a weight $\alpha_j$ to $h_j$ that is inversely proportional to the error rate. In this way best classifiers are considered more.
   - The weights for the next iteration, i.e. $w_{j+1}^i$, are reduced for the images $i$ that were correctly classified.
3. Set the final classifier to $h(\mathbf{x}) = \operatorname{sign}\left(\sum_{j=1}^{M} \alpha_j h_j(\mathbf{x})\right)$

**2.1.5 Classifier**

As we can see that the Viola-Jones algorithm scans the detector many times through the same image in order to see if it finds a match or not. Each time the detector or feature size is different. Even if an image should contain a face, it is obvious that a large number of sub-windows evaluated would yeild a negative result. The algorithm should spend time regions that have high chances of having a face and should disregard non-faces quickly. Hence, a single strong classifier that is a combination of all best features is not good to evaluate on each window because that would lead to a high computation 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 maybe a face. A given sub-window is immediately discarded as not a face if it fails in any of the stage.

Adaboost decides which classifiers or features to use in each stage.

**To make the algorithm efficient we must choose:**

Number of stages in cascade(strong classifiers). Number of features of each strong classifier And the threshold value of all the strong classifiers.

The problem here is that it is difficult to find an optimum combination. There was a solution suggested by Viola and Jones for Cascade Training.

**Manual Tweaking:**

Select fi (Maximum acceptable false positive rate)

Select di (Minimum acceptable true positive rate)

Select F target (Target overall false positive rate)

Until fi,di rates are met for this stage, keep adding features and train new strong classifier in Adaboost.

### 2.2 Advantages/Disadvantages

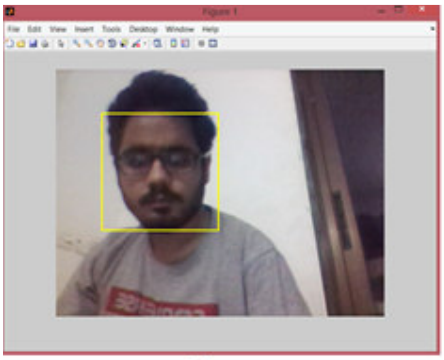
**Advantages**

- Fast features are computed very quickly.
- Feature selection is efficient.
- The features are scaled instead of scaling the image.
- This is a generic detection scheme which can be used to detect other objects like hands, buildings, etc.

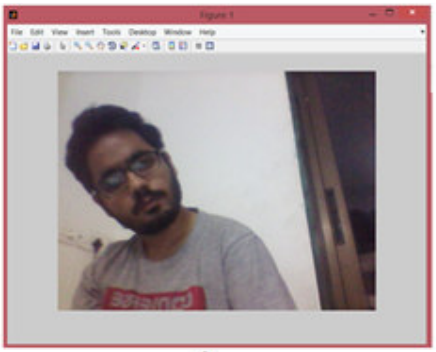
**Disadvantages**

- The detector is effective only in the case of frontal images of the face.
- If the face is turned 45 degrees, it fails the recognize the face.
- It is sensitive to lighting conditions
- Due to overlapping sub-windows, we might face the problem of multiple objects being detected as face.

**Detection Results**



1



2

As we can see in the above images that the face is detected only in the image 1 as it is the only image in which the face is looking directly at the camera and isn't tilted.

In image 2 the face is tilted hence the algorithm is unable to detect a face.

### 3. CamShift Algorithm

CAMshift [4] is an algorithm based on the mean shift algorithm and stands for Continuously Adaptive Mean Shift. It uses Hue channel to track objects. The Hue channel is based on HSV color model and objects of variety of colors can be recognized. After the CAMshift alogirhtm has fetched color information, it can track objects

faster and cosumes very less amount of CPU resources. As it consumes less computing resources, it has become one of the better real-time face tracking algorithms.

**CamShift Algorithm [4] steps:**

Set the region of interest (ROI) of the probability distribution image to the entire image.

Select an initial location of the Mean Shift search window. The selected location is the target distribution to be tracked.

Calculate a color probability distribution of the region centred at the Mean Shift search window.

Iterate Mean Shift algorithm to find the centroid of the probability image. Store the zeroth moment (distribution area) and centroid location.

For the following frame, center the search window at the mean location found in Step 4 and set the window size to a function of the zeroth moment. Go to Step 3.

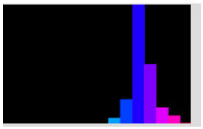
There are two important parts include in CAMShift procedure: histogram and search of peak probability.

In the first step of the procedure, the histogram of the tracked object is obtained. In the second step, the next frame will be converted into a map of skin color probability based on the histogram.

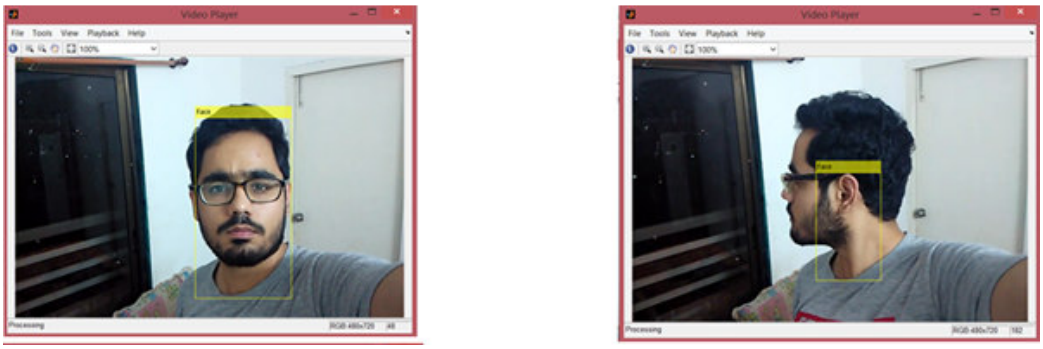
The peak probability centre is found in the third step, based on zero and first moment.

In the last step, the algorithm will check if it converges or not. If it does, it gets the position of the tracked object in this frame and fetches the next frame to track the object continuously, otherwise, it goes to step 3.

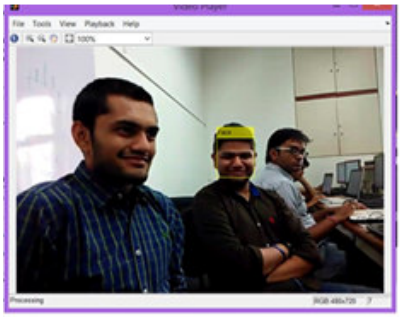
The histogram is the tracked object's color probability map. In the first step, the whole area of tracked object is scanned and the map which record how many pixels have a certain hue value in the tracked object area is built. And then, CAMshift finds out the peak the number of pixel in a certain hue value and normalizes the map into skin color probability map or histogram.

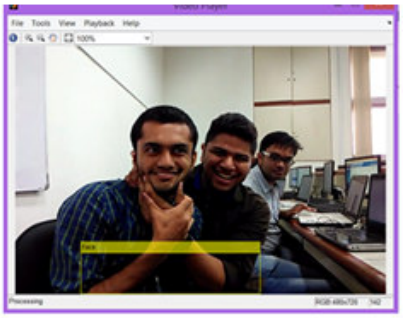


Detection results

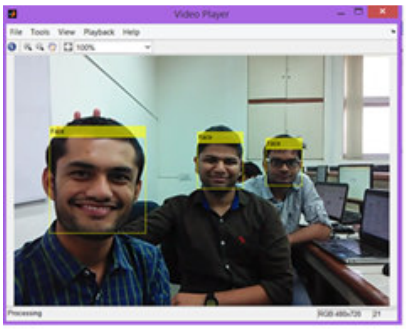


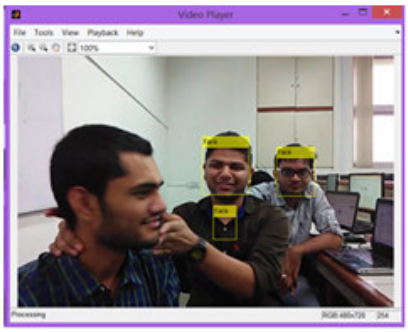
Viola-Jones + CamShift Algorithm Detection Test: Multiple Faces





Viola-Jones Algorithm Detection Test: Multiple Faces





**Observations**

Viola-Jones algorithm confused a shirt button as face when used for detecting and tracking multiple faces. Viola-Jones detects single face as well as multiple faces in a video as long as they are straight in front of the camera, looking upward without any tilt. CamShift algorithm tracked face properly for first few seconds and then started tracking hands thinking of them as face (probably because of skin color).

**Reference**


1. P. Viola, M.J. Jones, Robust Real-Time Face Detection, International Journal of Computer Vision, Vol. 57, No. 2, May 2004.
2. Rapid Object Detection using a Boosted Cascade of Simple Features, Paul Viola and Michael Jones, computer vision and pattern recognition, 2001
3. An Analysis of the Viola-Jones Face Detection Algorithm, Image Processing On Line, 2014–06–26.
4. Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces, John G. Allen, Richard Y. D. Xu, Jesse S. Jin http://crpit.com/confpapers/CRPITV36Allen.pdf
5. Robust real-time face detection, P Viola, MJ Jones - International journal of computer vision, 2004 – Springer
6. Fast and robust classification using asymmetric adaboost and a detector cascade, P Viola, M Jones - Advances in Neural Information Processing System, 2001