

# Multiple Face Detection and Tracking using Viola-Jones Algorithm

Anjali B Guptha<sup>1</sup>, Soorej K S<sup>2</sup>

<sup>1</sup>MTech Scholar, Dept. of ECE, Jawaharlal College of Engineering and Technology, Kerala, India

<sup>2</sup>Assistant Professor, Dept. of ECE, Jawaharlal College of Engineering and Technology, Kerala, India

\*\*\*

**Abstract** - Object detection and tracking are important tasks in computer vision applications such as surveillance, vehicle navigation, and autonomous robot navigation. Video surveillance in a dynamic environment, especially for human faces and vehicles, are current challenging research topics in computer vision. It is a technology to fight against terrorism, crime, public safety and for efficient management of traffic. The work involves designing of the efficient video surveillance system like searching of human life in complex environments like forest areas, wildlife parks, forest fires etc. Multiple Object Detection and Tracking are having a wide range of applications in real time video surveillance. By the usage of python the complex coding can be made simpler. Viola-Jones algorithm is used to develop the multiple face detection and tracking system. Its haar-features and cascade classifiers make it useful for the face detection and recognition.

**Key Words:** Viola-Jones algorithm, face detection, haar features, cascade classifiers

## 1. INTRODUCTION

Multiple Object Tracking (MOT) plays an important role in computer vision. The task of MOT is largely partitioned to, maintaining their identities, locating multiple objects and yielding their individual trajectories given in an input video. Objects to track can be, for example, pedestrians on the street, vehicles in the road, sport players on the court, or groups of animals (birds, bats, ants, fish, cells etc.). Multiple objects could be seen as different parts of a single object. As a mid-level task in computer vision, multiple object tracking grounds high-level tasks such as, behavior analysis, action recognition and, pose estimation. It has number of practical applications, such as virtual reality, human computer interaction and visual surveillance. In comparison with Single Object Tracking (SOT), which focuses on designing appearance models and motion models to deal with challenging factors such as out of plane rotations scale changes, illumination variations and multiple object tracking additionally requires two tasks to be solved: maintaining their identities, determining the number of objects, which typically varies over time. The common challenges that complicate MOT include among others: frequent occlusions, initialization and termination of tracks, similar appearance, interactions among multiple objects.

For decades the object detection in an image is improving as the image processing is developing through stages. Nowadays it's mainly developing through the software like MATLAB. Image processing is done to develop the video processing by detection of objects. The face detection and tracking of humans are really helpful during a difficult environment like forest fire, flood areas etc. The detection of an object can be done by using the CCTV images in parking areas, using cameras in the drones for searching of the human in dangerous situations. The focus of this project is on the face detection and tracking of that human. Here for demonstration we are using a webcam in which we are using a trained classifier for the identification of the human face by the position of the eyes, nose and mouth. Objects are mainly classified according to their size, shape, background, colour etc.

By using the Viola-Jones algorithm the facial features are analyzed. They give 93.24% of accuracy. It is done in MATLAB but in this project it is done using python language which is a powerful language in the technological world. It has an amazing ecosystem. There are standard library and it can be installed to do the needed function by easy coding. Its efficient high level data structures and a simple but effective approach to object oriented programming. Python's dynamic typing and its interpreted nature, elegant syntax, complete it as a perfect language for scripting and application development in many areas on most platforms. The Python interpreter and therefore the extensive standard library are freely available in source or binary form for all major platforms from the Python internet site. The Python interpreter is definitely extended with new functions and data types implemented in C or C++. Open CV is used for the computer vision application more easier. It has an interface with C++, python, java etc. It supports ios, window, linux, MacOS and android. The Dlib is also a library which is used to track the trained classifier. In face tracking detection of a group of faces in 1 frame of a video and establishing a connection between the following frames and keeping unique ID for each face. In each frame face detection is performed, a bounding box is drawn around the detected faces and displayed in the following frames. Dlib library has machine learning algorithms and tools that supports a wide range of domains including robotics and embedded devices. Correlation tracker from dlib library is used to track the object.

## 2. PROPOSED METHOD

### 2.1 Viola-Jones Algorithm

The system proposed here is one which can detect and track multiple faces with speed and accuracy. Opencv is used with Viola Jones algorithm to make it robust. The system built is able to detect faces that are in different angles, backgrounds, light effect etc from a video frame in real time.

A method used in face detection uses Haar like feature or better known as Haar Cascade Classifier. The main basis for object detection systems using the Haar Cascade Classifier is the classification system uses pixel intensity values and changes in contrast values between rectangular pixel groups that are close together. The contrast variance between pixel groups determines the white and black areas. Viola-Jones face detection procedure classifies images based on simple feature values. There are many reasons to use features rather than pixels directly. The most common reason is that features can be used to encode ad hoc domain knowledge that is difficult in the learning process with limited number of training data. Another reason for using features is the operating system based feature is much faster than pixel based systems. The Viola-Jones method is one of the most popular object detection methods. This method can provide results with a fairly high accuracy of about 93.7% with a speed of 15 times faster. To represent the image in Viola-Jones method Haar-like features are used. Viola-Jones method uses rectangular Haar-like features.

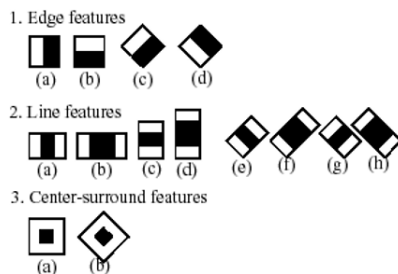


Fig 1: Common haar features

Haar is a long wave in the form of a square wave (one high interval and one low interval). In two dimensions, this square wave is depicted with a pair of squares that are adjacent to white and black. The presence of the features in any of the input image is determined by the Haar features. The result of each feature has been obtained by dividing the sum of black pixels by the sum of white pixels. Haar has an algorithm like this :

1. Each Haar feature consists of two or three rectangular sections that are colored "black" and "white".
2. The value of the Haar feature is the difference between the number of gray pixel values in the black and white rectangular area:

$$f(x) = \text{Sum of black rectangle (pixel gray level)} - \text{Sum of white rectangle (pixel gray level)}$$

3. Haar's features can reduce or increase the number of in-class or out-of-class variables, making classification easier. This is compared to raw pixel values.

4. The rectangle on the Haar feature can be calculated quickly using "Integral Image". The concept of integral images is used to calculate rectangles like features. The number of pixels in a rectangle is determined using the four values that exist in each corner of a particular rectangle. In the integral image, the value of each pixel is given by the number from the left and to the pixel above the image. All pixel values have been added in the D rectangle as shown in Figure 2 .

$$\begin{aligned} G1 &= A, \\ G2 &= A+B, \\ G3 &= A+C, \\ G4 &= A+B+C+D \\ G1+G4-G2-G3 &= A+A+B+C+D-A-B-A-C \\ &= D \dots\dots\dots(1) \end{aligned}$$



Fig 2. Integral Image

Image Integration is used in determining hundreds of Haar features in an image and used on different scales. Integration is the addition of small units in the form of pixel values simultaneously. The integral value of each pixel is the sum of all pixels from top to bottom, from top left to bottom right. The whole image can be summed using several integer operations per pixel. Selection of specific features will be used to set the threshold value. Adaboost is one of the boosting algorithms which functions to select large numbers of features that only choose certain features. This is done by evaluating each feature of the training data by using the value of the feature. Features that have the greatest restrictions between objects and non-objects are considered the best features. AdaBoost Learning can also be used to improve classification performance with simple learning to combine many weak classifiers into one powerful classifier. The next step is combine a complex classifier in a multilevel structure that can increase the speed of object detection by focusing on the area of the image that has a chance.

The face object is determined using Viola Jones which refers to the flow where the grayscale image will be scanned per-sub-window to look for positive features with AdaBoost and Cascade Classifier. In the Viola Jones the face detection algorithm is controlled by special trained scanning window classifiers. Face detection can be seen as a problem of

pattern classification with two classes, namely "face" and "non face". Non-face class consists of images that can explain something that is not a "face", while the class "face" contains or consists of all face images. To detect faces in the input image, all images are scanned and images of areas identified as faces or non faces.

### 2.2 Haar Cascades

Haar Cascades use machine learning techniques during which a function is trained from tons of positive and negative images. This process within the algorithm is feature extraction. The training data utilized in this project is an XML file called:

haarcascade\_frontalface\_default.xml

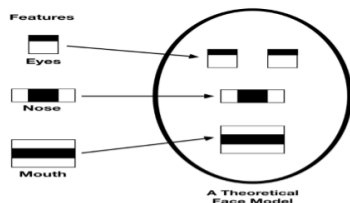


Fig 3: Features of face

The photos were either taken at a public event or from my photography portfolio. For uniformity in testing, choose a picture resolution size of 500 x 331 pixels in JPEG format for every photo tested. The aim of the test is to ascertain the accuracy in face detection employing a sort of samples. We tend to urge higher accuracy for a more specific sort of image, where the location of the face is extremely detectable and ideal. Instead use various images where the face isn't always located dead centre or ones where there's a gaggle of individuals (more than 1 face in an image). We are getting to use the detect Multiscale module from OpenCV. This create a rectangle with coordinates (x,y,w,h) round the face detected within the image.

### 2.3 Adaboost': To Improve Classifier Accuracy

As acknowledged above, quite 180,000 features values result within a 24X24 window. To only select the simplest feature out of the whole chunk, a machine learning algorithm called Adaboost is employed . What it essentially does is that it selects only those features that help to enhance the classifier accuracy. It does so by constructing a robust classifier which may be a linear combination of variety of weak classifiers. This reduces the quantity of features drastically to around 6000 from around 180,000.

### 2.4 Cascade of Classifiers

The cascade classifier essentially consists of stages where each stage consists of a robust classifier. this is often beneficial since it eliminates the necessity to use all features directly on a window. Rather, it groups the features into

separate sub-windows and therefore the classifier at each stage determines whether or not the sub-window may be a face. just in case it's not, the sub-window is discarded along side the features therein window. If the sub-window moves past the classifier, it continues to subsequent stage where the second stage of features is applied. the method are often understood with the assistance of the diagram below.

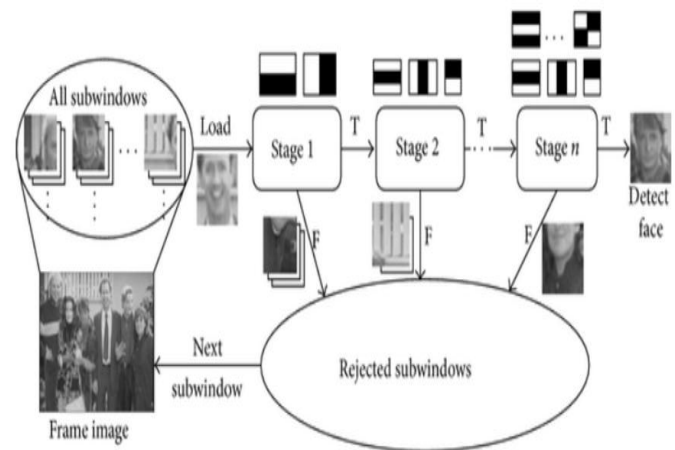


Fig 4: Cascade structure of haar classifier

### 2.5 Block Diagram of Face Detection

- STEP 1: Acquire new image from camera
- STEP 2: convert the colour image to gray scale image
- STEP 3: detect faces using trained cascade classifiers
- STEP 4: detect eyes
- STEP 5: normalize the face images and orientation
- STEP 6: facial features are recognized by the haar cascade classifier
- STEP 7: the faces are detected as per the classifiers trained

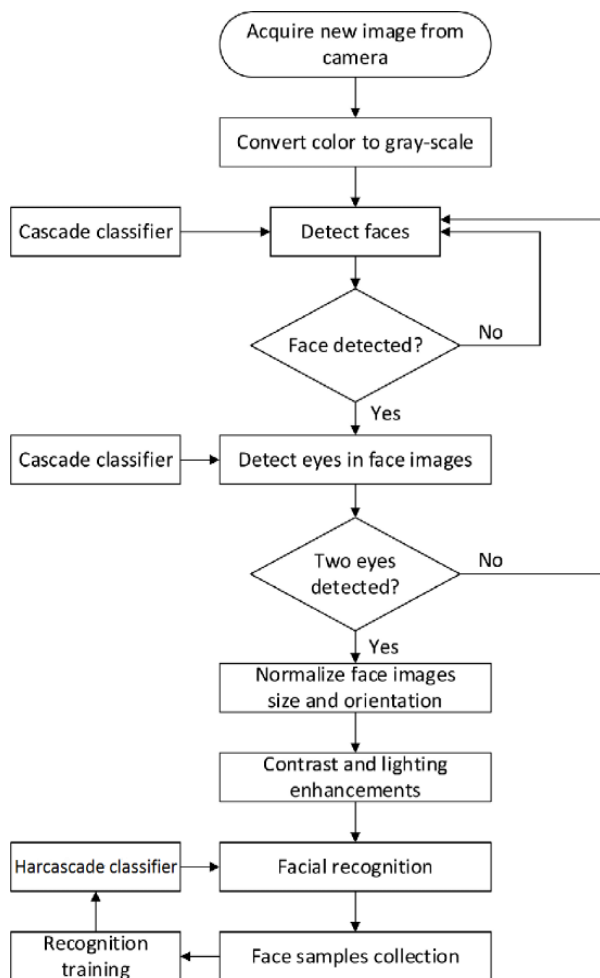


Fig 5: face detection

## 2.6 Tracking of Faces

### 1.OpenCV Installation

- Install Python 3.7
- Install Numpy
- Download the latest version of OpenCV in Sourceforce or GitHub
- Extract the OpenCV. From the folder where you extracted, goto folder: yourOpenCVFolder\opencv\build\python\3.7
- Copy file cv2.pyd to your python folder\lib\site-packages.

2.Basic Video Capture: Capture the video using the webcam on your laptop. It will open a window contains the images captured by the webcam. Press q to quit the window.

### 3.Saving Video

4.Loading and Playing Video:Now load and play the video that we just captured.

5.Change Window Size:The size of the window can be changed by setting the frame size (e.g. 640x480; 320x240; 960x720), larger is slower ad you can see the delay from your movement.

6.Color Transformation: Use basic video capture code. Now we have a simple color transformation

7.Image Enhancement:Using Histogram equalization, we can enhance the contrast of the image. OpenCV equalizeHist() function is works for grayscale image. The following code is useful to enhance the constrast of color image.

8.Image Transformation

9.Edge Detection & Smoothing:Use Canny Edge detection .

10.Super Impose:Use the edge detection as our mask and superimpose it with the original image. First we reverse the edge detection result using bit wise Not to inverse it. Then, we use bit wise And to superimpose with the blur image.

11.Thresholding: Use threshold to create a mask. A mask is a binary image.

Adaptive threshold has the following parameters:

- src – Source 8-bit single-channel image.
- dst – Destination image of the same size and the same type as src .
- maxValue – Non-zero value assigned to the pixels for which the condition is satisfied. Put 255.
- adaptiveMethod – Adaptive thresholding algorithm to use,
- ADAPTIVE\_THRESH\_MEAN\_C or ADAPTIVE\_THRESH\_GAUSSIAN\_C .
- thresholdType – Thresholding type that must be either THRESH\_BINARY or THRESH\_BINARY\_INV .
- blockSize – Size of a pixel neighborhood that is used to calculate a threshold value for the pixel: 3, 5, 7, and so on.
- C – Constant subtracted from the mean or weighted mean.

12.Countour: Two main functions of contours are :

- img,contours,hierarchy= cv2.findContours(thresh,cv2.RETR\_TREE,chainRule Approximation)
- img=cv2.drawContours(img, contours, index, colorTuple, thickness)

The arguments are:

- img = image
- index=-1 means show all contours, 0-len(contours) means to show each contour
- chainRuleApproximation is either:
- cv2.CHAIN\_APPROX\_SIMPLE: to give only 4 points in a rectangle

- cv2.CHAIN\_APPROX\_NONE: to give all points  $\square$  colorTuple is any BGR color. For instance:
- (255,0,0) for Blue
- (0,255,0) for Green
- (0,0,255) for Red
- thickness is integer 1-10

13.Optical Flow

14.Image Difference: Motion Detection

Image difference from one frame to another is a way to detect the moving object. Two images are captured with a slight time delay of 1/25 seconds. The image difference, get the contour out of it and put the bounding box out of the contour.

15.Face Detection

16.Tracking using coorelation tracker from dlib library.This lets to trace the position of an object because it moves from frame to border during a video sequence. To use it, you give the correlation\_tracker the bounding box of the thing you would like to trace within the current video frame. Then it'll identify the situation of the thing in subsequent frames.

**3. RESULT AND ANALYSIS**

The first step in detecting a face is the process of finding facial shapes from the nose, ears, right eye, mouth using the cascade classifier. The next step is AdaBoost Learning and Integral Image. AdaBoost learning strengthens the face shape of the previous process with a large and accurate detection number. This is because the previous detection has obtained initial face data. Next is the Integral Image process. Processing the haar feature will form a square pattern around the face pattern that has been processed before. The haar feature must ensure that the face data provided in the previous process is valid. This process adjusts the position of the rectangular box on the face pattern in the face area Face detection accuracy testing is done by inputting several images.

In frame 81 the girl has bend down her head and both of the faces of the girls are detected by the trained classifier by checking the features of the face.

In the frame 85 the girl is standing up and both of the faces are tracked. Then the rectangular bounding boxes shows the tracked portion on the image.

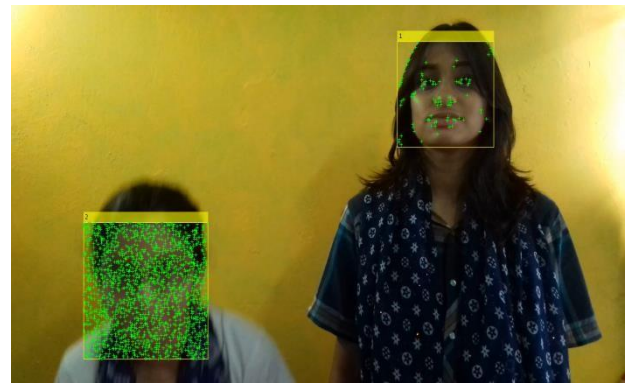


Fig- 6:Frame 81

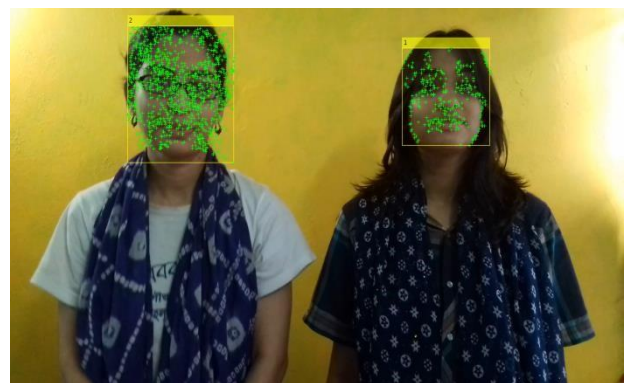


Fig 7:Frame 85

**4. CONCLUSION**

Face detection and tracking system is a part of image processing applications and its importance as a research area is increasing recently. Implementations of system are video surveillance, human rescue and similar security activities. The main goal of the thesis is to implement a face detection and tracking system which is useful to detect and track human faces in difficult environments. The goal is reached by using face detection and tracking methods. Facial feature face detection methods are used to find position and extract faces in achieved images for frames of video. Implanted methods are Viola Jones algorithm. Also, dlib library from python is used for face tracking. A haar cascade classifier is employed to classify and solve face detection problem.

**ACKNOWLEDGEMENT**

I am ineffably indebted to my guide Mr Soorej K S for his active guidance and encouragement to accomplish this assignment and Wikipedia for the correct source of information.

## REFERENCES

- [1] Souhail Guennouni, Anass Mansori and Ahaitouf Ali, 2015. Multiple object detection using OpenCv on an Embedded platform: IEEE Transactions on Image Processing
- [2] Panth Shah, Tithi Vyas, 2014. Interfacing Of Matlab With Arduino For Object Detection Algorithm Implementation Using Serial Communication: IEEE transactions on Image Processing.
- [3] Hrishikesh Kulkarni, Dr. Ghassem Tofighi (2018), Unconstrained Facial Recognition using Supervised Deep Learning On Video: Ryerson University
- [4] Afef Salhi, Yacine Moresly, Fahmi Ghazzi (2017). Modeling From An Object And MultiObject Tracking System: Global Summit On Computer & Information Technology 2017 IEEE.
- [5] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, Xiaowei Zhao and Tae-Kyun Kim (2017). Multiple Object Tracking: A Literature Review
- [6] W. Hu, T. Tan, L. Wang, and S. Maybank (2004); A survey on visual surveillance of object motion and behaviors: IEEE Trans. Syst. Man Cybern.