

# FACE DETECTION AND TRACKING ALGORITHM USING OPEN CV WITH RASPBERRY PI

Savitha M <sup>1</sup>, Divya<sup>2</sup>

<sup>1</sup>Asst.Professor, Dept. of Computer Science and Engineering, Vivekananda College of Engineering and Technology, Puttur, D.K, Karnataka

<sup>2</sup>Asst.Professor, Dept. of Computer Science and Engineering, Vivekananda College of Engineering and Technology, Puttur, D.K, Karnataka

\*\*\*

**Abstract** - This paper mainly describes the proposed model, face detection and tracking using OpenCV software interfacing with Raspberry pi. The aim of this paper is to develop a real-time application like security system that is necessary in several platforms. In this paper the real-time face detection and tracking is implemented using hardware devices like pi cam and Raspberry pi. We present a face detection and tracking algorithm in real time camera input environment. The entire face tracking algorithm is divided into two modules. The first module is face detection and second is face tracking. To detect and track the face in the live video stream, Haar based algorithm is used. The algorithm is applied on the real-time camera input and under real time environment conditions.

Key Words: **Face detection, Face tracking, Haar-Cascade algorithm**

## 1. INTRODUCTION

With the number of camera installed all around the world steadily growing, real time analysis of video content is becoming increasingly important. Among objects of possible interest, faces have received significant attention both in academia and in industry. The accurate localization of faces in a video is necessary first step to initialize other computer vision tasks, such as face recognition. The computer vision human face detection is an important research topic. It is needed for many computer applications like HCL, surveillance, human-robot interaction etc. In this field facial tracking is finding more growth of use in security and safety applications to detect various situations. Detecting human faces in a live video is a great challenging problem. For face detection and tracking on a live stream video different algorithm have been introduced. Each algorithm has got its own advantages and disadvantages. But any face tracking Algorithm will have some errors which will cause deviation from required object. The tracker can be accurate if and only if it is able to minimize this deviation. The technique used is one of the effective approaches. It is quicker and simple. Haar Cascade is a machine learning object detection algorithm used to identify objects in an image or video and based on the concept of features. Video processing has become a major requirement in current world. This technique is majorly used to detect and track human faces. Face detection and tracking is the phase where

we detect a person's face from a video sequence and track him/her through the video. It plays a vital role in video corrections, surveillance and military tracking so on. Detecting a face is a computer technology which let us know the locations and size of human faces. This helps in getting facial features and avoiding other objects and things. Face detection is preliminary step for many other applications face recognition, video surveillance etc.

## 2. RELATED WORK

Chathrath and Gupta et. al[5] proposed a method of detecting the face in an image. This paper presents a set of detailed experiments on difficult face detection and tracking data set which has been widely studied. This data set includes faces under a wide range of conditions including: illumination, scale, pose and camera variation. Divya George and Arunkant [6] proposed a method for face detection and tracking in a video using optical flow techniques. The tracking is implemented using different methods of optical flow such as block based methods, discrete optimization methods, Differential methods. In the work, Pyramidal Lucas-Kanade Feature Tracker optical flow method is used for tracking the face. Z. Liu and Y. Wang [8] proposed a method for face detection and tracking. The Algorithm is based on template matching based on dynamic Programming. The algorithm is based on template matching based on dynamic programming. Preliminary experimental results show that the face detection algorithm is fast as well as reliable, and the face tracking and clustering method are promising.

## 3. PROPOSED ALGORITHM

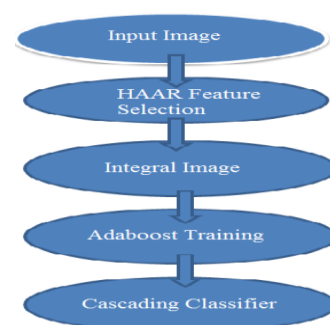


Fig -1: Flow chart of Haar Cascade algorithm

The algorithm used is Haar Cascade is a machine learning object detection algorithm used to identify objects in an image or video based. The above figure 4.2 shows the flow chart of Haar Cascade. Haar Cascade is trained by superimposing the positive image over a set of negative images. The training is generally done on a server and on various stages. The Characteristics of the algorithm are:

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

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

### 3.1 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.

### 3.2 Disadvantages

- The detector is effective only in the case of frontal images of the face.
- If the face is turned 45 degrees, it fails to 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.

## 4. METHODOLOGY

The algorithm comprises of four stages:

1. Haar Features Selection
2. Creating Integral Image
3. Adaboost Training Algorithm
4. Cascade Classifiers

### 4.1 Haar Features

Haar-like features are digital image features used in object detection. 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.

The black region in a Haar feature is replaced by +1 and white region is replaced by -1. Viola Jones algorithm[7] 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.

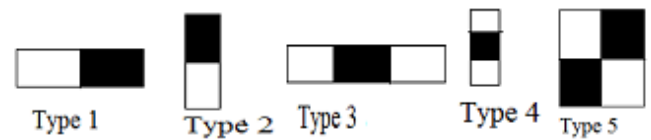


Fig -2: Haar features used in viola-jones

### 4.2 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. 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. In an integral image the value at pixel(x, y) is the sum of pixels above and to the left of (x, y).

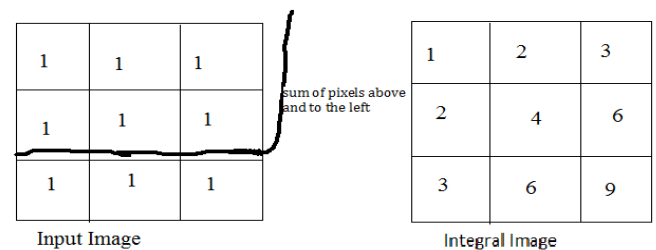


Fig -3: Input and Integral image

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

### 4.3 Adaboost

Adaboost 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. Adaboost constructs a strong classifier as a linear combination of these weak classifiers.

In a standard 24x24 pixel sub-window, there are a 162,336 total 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.

#### 4.4 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 yield 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.

#### 4.5 Tracking of Face

Face tracking has been an important and active research field because it offers many applications, especially in video surveillance, biometrics, or video coding. The goal of this project was to implement a real-time system using Open CV and webcam d to detect and track a human's face. The face detection algorithm involved color-based skin segmentation and image filtering. The face location was determined by calculating the centroid of the detected region. A software version of the algorithm was independently implemented and tested on still pictures in Open CV.

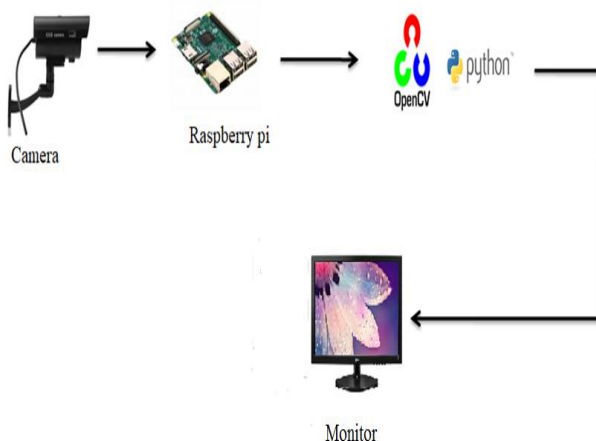


Fig -4: Tracking of face

#### 4. SIMULATION AND RESULTS

The Figure 5 shows the detection of the face using Open CV and Raspberry.



Fig -5: Detection of Face

Figure 6 shows the setup for the implementation of the face detection and tracking using Open CV and raspberry pi.



Fig -5: Hardware Set-up

#### 5. CONCLUSION

Multi-face detection and tracking in a live video is a very crucial topic of surveillance systems in field applications. We implemented the different detecting and tracking methods. Algorithms work well for detection and tracking. The Haar-features gives the most useful results for our project. The approach presented here for face detection and tracking decreases the computation time producing results with high accuracy. Tracking of a face in a video sequence is done using Haar-Cascade algorithm is used for detection facial features. Using this system many security and surveillance system can be developed.

#### REFERENCES

[1] Ayi, Maneesh, Ajay Kamal Ganti, Maheswari Adimulam, and Badiganti Karthik. "Interfacing of MATLAB with Arduino for face detection and tracking algorithm using serial communication." In 2017 International Conference on Inventive Computing and Informatics (ICICI), pp. 944-948. IEEE, 2017.

- [2] Horak, K., and L. Zalud. "Image Processing on Raspberry Pi in Matlab." *Advances in intelligent systems and computing* 4 (2015).
- [3] Comaniciu, Dorin, and Visvanathan Ramesh. "Robust detection and tracking of human faces with an active camera." In *Proceedings Third IEEE International Workshop on Visual Surveillance*, pp. 11-18. IEEE, 2000.
- [4] Yan, Chen, Zhengqun Wang, and Chunlin Xu. "Gentle Adaboost algorithm based on multi-feature fusion for face detection." *The Journal of Engineering* 2019, no. 15 (2019): 609-612.
- [5] J. Chatrath, P. Gupta, P. Ahuja, A. Goel and S. M. Arora, "Real Time Human Face Detection and Tracking", *International Conference on Signal Processing and Integrated Networks (SPIN)*, (2014), 20-21 February, Noida.
- [6] Divya George, Arunkant A. Jose, "Face Detection And Tracking At Different Angles In Video Using Optical Flow", *ARPN Journal of Engineering and Applied Sciences* Vol. 10, No. 17, September 2015.
- [7] P. Viola and M. Jones, "Robust real-time face detection", *International journal of computer vision*, vol. 57, no. 2, (2004), pp. 137-154.
- [8] Z. Liu and Y. Wang, "Face Detection and Tracking in Video Using Dynamic Programming" *Proc. Int'l Conf. Image Processing*, 2000.