



Automated Attendance System

¹Pallikonda Lokesh, ¹Shaik Ibrahim, ¹Gaddam Thanmayee Reddy, ²Srinivas Y

¹Student, Department of Computer Science and Engineering, GITAM University, Visakhapatnam, Andhra Pradesh, 530045, India.

²Professor, Department of Computer Science and Engineering, GITAM University, Visakhapatnam, Andhra Pradesh, 530045, India.

Abstract : With the excessive use of monitoring cameras installed in public places, schools, colleges, hospitals, and homes, an application for tracking and detecting faces in videos used for multipurpose activities is vital. So this real-time application detects, recognizes the faces, and updates the attendance of the students. For that, we are using an open-source computer vision called OpenCV. This project needs a face tracking framework capable of face detection using Adaboost and Haar cascades. To achieve Face recognition from video, image processing techniques are used. This application runs efficiently and smoothly to automatically mark the attendance and track multiple persons in real-time.

I.INTRODUCTION

A face recognition system automatically recognizes a person from an image or a video source. The recognition task is by obtaining facial features from a picture of the subject's face. The main objective of video-based face recognition and tracking is to identify a video face track of unknown individuals. It identifies facial features by extracting features from an image of the subject's face and analyzing the relative position of eyes, size, shape, etc. These obtained features are to search corresponding matching features from the photos in the database to display related person details.

In brief, There are three steps in a face recognition system: face detection, feature extraction, and comparing faces. Face detection is to identify faces in the video, and then facial recognition is done to match the detected faces with the faces in the database. For this project, we use technologies like Sublime Text, Visual studio, OpenCV, HAAR Cascades, and Adaboost.

The most popular algorithm for this is the Viola-Jones algorithm based on a cascaded Haar feature detector. Viola and Jones give it in their seminal 2001 publication, Rapid Object Detection using a Boosted Cascade of Simple Features. OpenCV is the most popular object detection algorithm. One of the primary benefits of Haar cascades is that they are just so fast.

"Sublime Text" is one of the best text editors available in the market for coding and implementation. Here, we used Visual Studio to install some extensions and modules regarding Python in our project. Windows needs Visual Studio to add Python extensions like CMake, Face-Recognition, and Dlib.

II. METHODOLOGY:

Flow-chart:

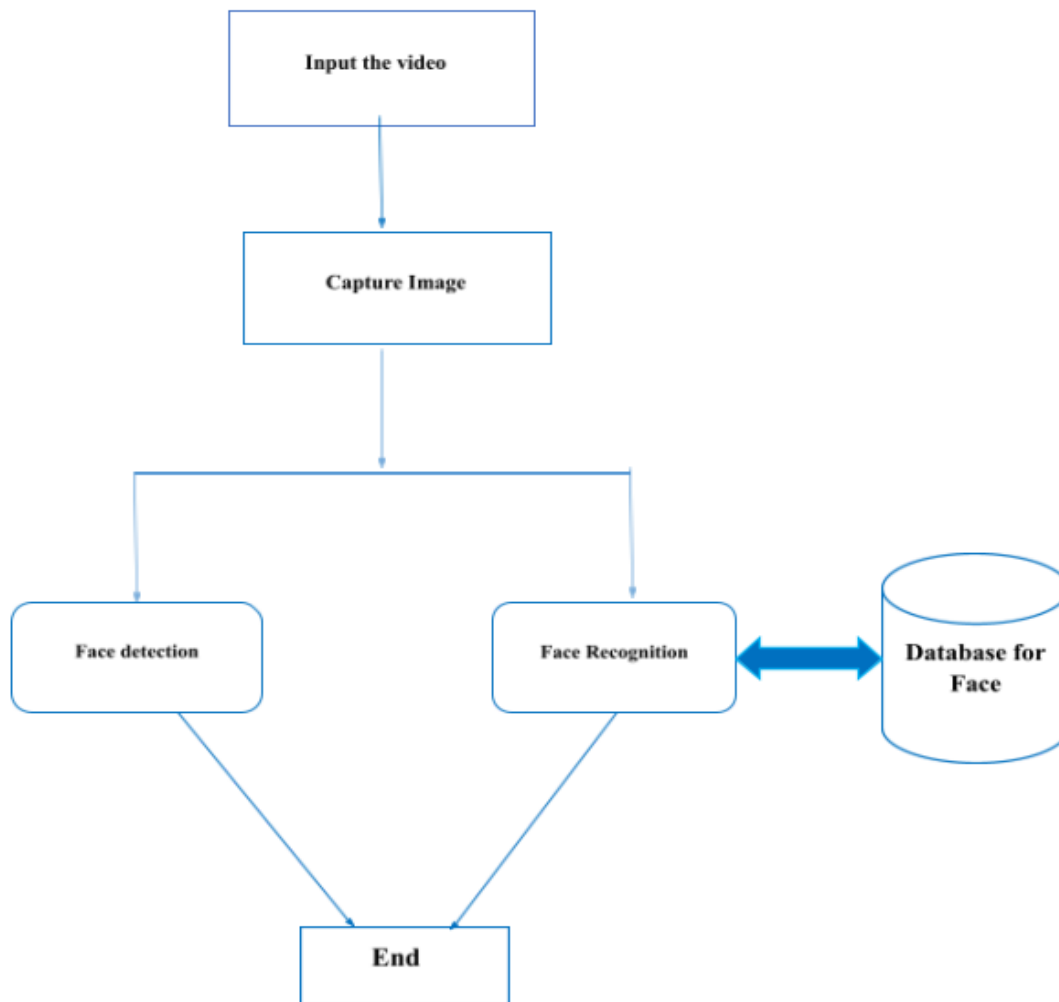


Fig-1: Flow chart

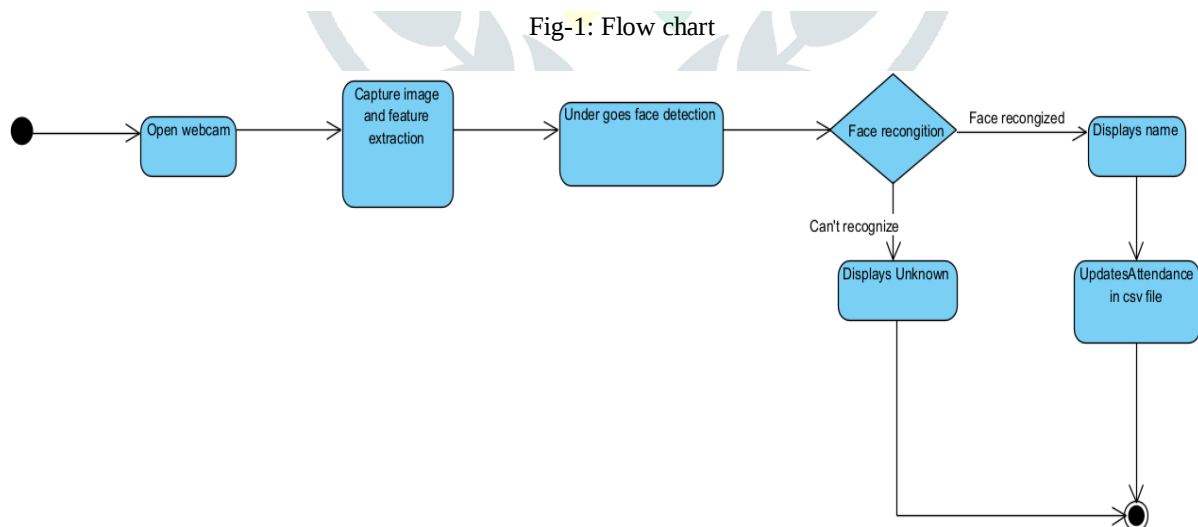


Fig-2:State machine diagram

Methodology:

We need to input a video, then a rectangular frame captures the faces of a person, and then haar cascade classifiers and AdaBoost are used to detect and recognize the faces. The recognized faces are checked with the database of the image. If a match is found, then it displays the known person's name and marks the attendance. If an intruder is found, then it shows unknown. If a new student or any authorized person needs to be detected, we need to keep the record of their image in the database.

The purpose of face detection implies that we discover the faces from the whole picture or recordings or real-time video of a person. By using the Haar cascade classifier, the faces get detected.

There are four steps of the Haar Cascade classifier:

Haar features selection

Creating an integral image

Adaboost training

Cascading classifiers

This algorithm requires several positive and negative photos to identify faces. Good photographs of a mask and bad images without a look are positive images. It requires that these photos train the classifier. It determines each domain's aggregate pixel strength and seeks the truths within these totals. Adaboost is then used to detect the best element, and Adaboost prepares the classifier for it.

Face-Recognition:

Face Detection: The first task we perform is detecting faces in the image or video stream. Now that we know the exact coordinates of the faces, we extract these faces for further processing.

Feature Extraction: We have cropped the face from the image, and we extract features. Now we will use face embeddings to remove the elements from the face. A neural network takes an image of the person's face as input and outputs a vector representing the most critical features of a look. In machine learning, we call this vector face embedding.

Comparing faces: Now that we have face embeddings for every look in our data saved in a file, the next step is to recognize a new image that is not in our data. So the first step is to compute the face embedding for the image using the same network we used above and then compare this embedding with the rest of the embeddings we have.

We can now install the "Face-Recognition" pack using the pip command as we've seen what we can do with Face-Recognition".

First, like OpenCV, the Face-recognition extension also needs some essential tools like CMake and Dlib to run some of the functions we use in the project source code.

As we already know, CMake and Dlib were already installed when we installed "Developing C/C++" in Visual Studio. We need to add CMake to the Environmental Variables when we can access it across the Windows OS.

We can visit "CMake.org" and download the windows installer, automatically adding CMake to the path, or downloading the Zip file for manual installation.

After adding CMake to the Environment PATH, we can install Face-Recognition without any errors.

We use the same pip command to install Face-recognition in Windows using the Windows PowerShell or Command Prompt.

The command "pip install face-recognition" will install Face-Recognition in the system and upgrade CMake and Dlib if any upgrades exist.

We have all the extensions and modules we need for the project's source code. It's time to add HAAR CASCADES.

HAAR CASCADE: Viola and Jones, in their seminal 2001 publication, Rapid Object Detection uses a Boosted Cascade of Simple Features, arguably OpenCV's most famous object detection algorithm. One of the primary pros of Haar cascades is that they are too fast; it's tough to beat them in terms of their speed.

The disadvantage to Haar cascades is that they tend to be prone to false-positive detection, require parameter tuning when being applied for detection, and, in general, are not as accurate as the more "modern" algorithms we have today.

That said, Haar cascades are

1. An essential part of the computer vision and image processing literature
2. Still used with OpenCV

3. Still helpful, mainly when working in resource-constrained devices when we cannot afford to use more computationally expensive object detectors.

As haar cascades are open source in-built files, they are available online. Some best ways to acquire haar cascades are by looking into some GitHub repositories and websites that provide the folders containing the haar cascade files.

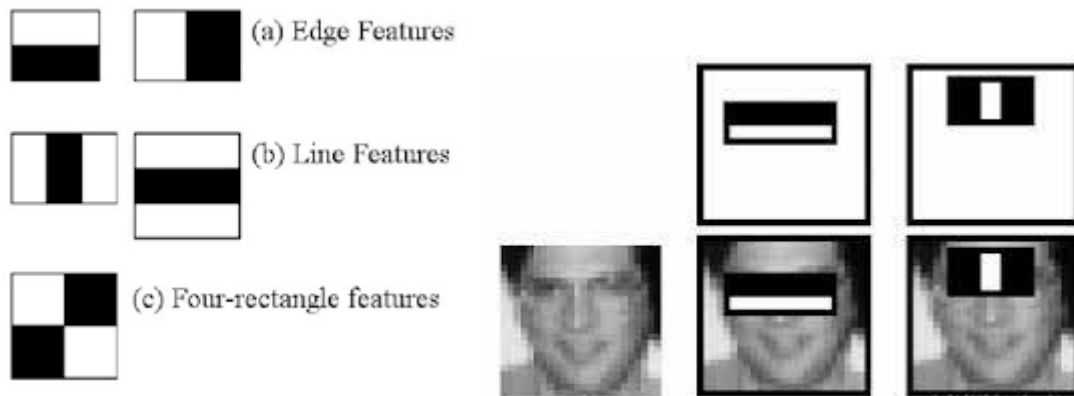


Fig-3: Haar Features

The goal is to calculate the total of all pixel intensities in the dark part of the haar feature and the total of all pixel intensities in the light portion of the haar feature. Then figure out how they vary. The haar value will be closer to 1 if the photo has a boundary dividing dark pixels on the right from light pixels on the left.

The very first pair of two rectangular characteristics is in charge of determining whether the edges are horizontal or vertical (as shown above). The next set of three rectangular characteristics determines whether a brighter zone is flanked on either side by darker sections or conversely. The final stage of four rectangular characteristics is in order of determining how intensity values fluctuate across diagonals.

III. RESULTS AND DISCUSSIONS

Output:

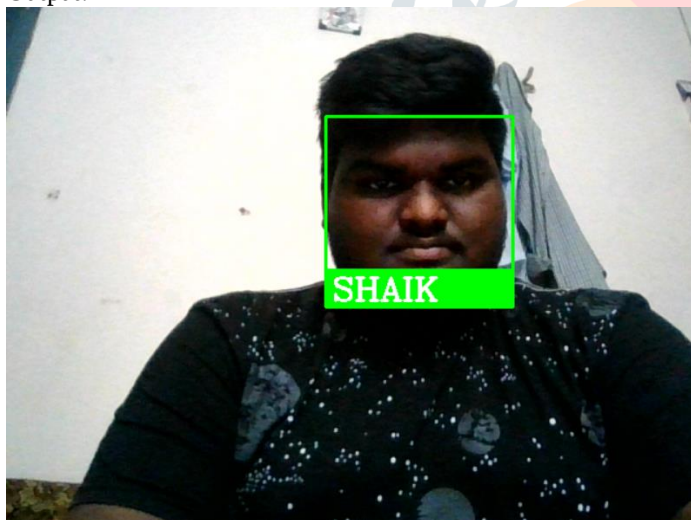


Fig-4: Camera view

When the attendance.py file is executed, the webcam captures the face, detects it, and recognizes it. Then it updates the CSV file like:

```
1 Name, Time
2 shaik, 13:51:23
```

Fig-5: Output

It displays the name and time of the student in the attendance.csv file.

The attendance management system is straightforward and effective. Face recognition-based Automated Attendance Systems are both time-saving and safe. Shortly, the efficiency might be enhanced by incorporating additional approaches. We want to use and control the interaction between our system, users, and administrators in future work to increase face recognition efficacy. On the other hand, our approach could be used in an entirely new aspect of face recognition applications, mobile-based face recognition, which can help ordinary people learn about anyone who is photographed by a smartphone camera.

IV. CONCLUSION & FUTURE SCOPE

Our primary point is to identify and mark the attendance of the student. Human face detection is in different fields, such as law enforcement, equity arrangements, identification recovery, and Biometrics. Facial identity and recognition technology integrate university campus tech solutions to ensure student protection. We establish facial recognition based on face detection for chosen locations, such as university campuses. We are eager to develop a quick and efficient facial recognition framework that identifies and does not reach people's faces.

Nowadays, we are not moving out without face masks, so we want to introduce a feature in our application that identifies a person in the future.

By calculating the illegal movement of outsiders using a suggested methodology, we will minimize internal campus crimes much as our point is to expand transparent mindfulness, human protection, and authorization of the law by adopting our proposal

REFERENCES

- [1]-SREEKANTHA., VIJAYALAKSHMI S, HARI PRASAD, HRITHIK, NAGAJYOTHI M S, RAKSHITHA S: FACE-BASED CCTV ATTENDANCE MONITORING SYSTEM USING DEEP FACE RECOGNITION. IRJET 6th June 2021
- [2]-Evangelos Michos, Stamatios Gkoumas, Ioanna Siakampeti, Christos Fidas, University of Patras, Greece: On the Extension of the Haar Cascade Algorithm for Face Recognition: Case Study and Results. ACM November 20–22, 2020
- [3]-Zankruti Arya, Vibha Tiwari, ECE Department Medi-Caps University Indore (M.P.), India: Automatic Face Recognition and Detection Using OpenCV, Haar Cascade and Recognizer for Frontal Face. International Journal of Engineering Research and Applications, 6th June 2020 Problem
- [4] C. Panjaitan, A. Silaban, M. Napitupulu, J.W. Simatupang, "Comparison K-Nearest Neighbors (K-NN) and Artificial Neural Network (ANN) in Real-Time Entrants Recognition". 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI).
- [5]D. T. P. Hapsari, C. G. Berliana, P. Winda, M. A. Soeleman, "Face Detection Using Haar Cascade in Difference Illumination." 2018 International Seminar on Application for Technology of Information and Communication.
- [6]N. A. Othman, I. Aydin, "A face recognition method in the internet of things for security applications in smart homes and cities." 2018 6th International Istanbul Smart Grids and Cities Congress and Fair (ICSG).
- [7]Simpson E A, Maylott S E, Leonard K, Lazo R J and Jakobsen K V 2019 Face detection in infants and adults: Effects of orientation and color Journal of experimental child psychology 186 17- 32.
- [8]Manoharan, S. (2019)," Smart Image Processing Algorithm For Text Recognition, Information Extraction And Vocalization For The Visually Challenged," Journal of Innovative Image Processing (JIIP), 1(01), 31-38.
- [9]Ashu Kumar, Amandeep Kaur, Munish Kumar, "Face detection techniques: a review" Springer Nature B.V. (2018) DOI: 10.1007/s10462-018-9650-2 3.
- [10] Illias Maglogiannis, Demosthenes Vouyioukas, Chris Aggelopoulos, "Face detection and recognition of natural human emotion using Markov random fields." DOI 10.007/s00779 007 0165 0.