FACE ATTENDANCE

Submitted by:

ADITYA AGARWAL (191B020)
ASHMIT MODI (191B076)
ANSHU GUPTA (191B052)

Name of Supervisor- Mr.Navaljeet Singh Arora

Submitted in partial fulfillment of the Degree of Bachelor of Technology

Department of Computer Science & Engineering

Feb 2022- Jun 2022

JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY,

A-B ROAD, RAGHOGARH, DT. GUNA - 473226, M.P., INDIA

DECLARATION

We hereby declare that the work reported in 5th semester Minor project entitled "FACE ATTENDANCE", in partial fulfillment for the award of the degree of B.Tech (CSE) submitted at Jaypee University of Engineering and Technology, Guna, as per the best of our knowledge and belief there is no infringement of intellectual property rights and copyright. In case of any violation, we will solely be responsible.

Aditya Agarwal (191B020)

Anshu Gupta (191B052)

Ashmit Modi (191B076)

Jaypee University of Engineering and Technology,
Raghogarh, Guna – 473226
Date: 3/6/2022

CERTIFICATE

This is to certify that the project titled “**FACE ATTENDANCE**” is the bona fide work carried out by **Aditya Agarwal ,Ashmit Modi** and **Anshu Gupta**, a student of B Tech (CSE) of Jaypee University of Engineering and Technology, Guna (M.P) during the academic year 2020-21, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (Computer Science and Engineering) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar tile.

Signature of the Guide

Jaypee University of Engineering and Technology, Raghogarh,
Guna – 473226

Date: 3/6/2022

ABSTRACT

Face detection and face recognition are very important technologies these days, furthermore we noticed that they got have a variety of uses such as cellphones, army uses, and some high risk information offices. We decided to make a device that detects and recognize the face as a student attendance system and can be a substitute for the regular paper attendance system and finger print attendance system. This project was an attempt at developing an Face detection and tracking system using modern computer vision technology. Office Excel sheet to include students names, and a computer (or laptop) to integrate the programs together. The results of the project are expressed in this report, and amount to the application of computer vision techniques in tracking animate face in both a 2 dimensional and 3 dimensional scene.

ACKNOWLEDGEMENT

We would like to express our gratitude and appreciation to all those who gave us the opportunity to complete this project. Special thanks is due to our supervisor **Mr. Navaljeet Singh Arora** whose help, stimulating suggestions and encouragement helped us in all the time of development process and in writing this report. We also sincerely thanks for the time spent proofreading and correcting my many mistakes. We would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited period. Last but not the least I am grateful to all the team members of **FACE ATTENDANCE.**

Thanking you

Aditya Agarwal (191B020)

Anshu Gupta (191B052)

Ashmit Modi (191B076)

LIST OF FIGURES

Table of Contents

CHAPTER-1
INTRODUCTION

1.1 Problem Definition

Traditional student attendance marking technique is often facing a lot of trouble. The face recognition student attendance system emphasizes its simplicity by eliminating classical student attendance marking technique such as calling student names or checking respective identification cards. There are not only disturbing the teaching process but also causes distraction for students during exam sessions. Apart from calling names, attendance sheet is passed around the classroom during the lecture sessions. The lecture class especially the class with a large number of students might find it difficult to have the attendance sheet being passed around the class. Thus, face recognition student attendance system is proposed in order to replace the manual signing of the presence of students which are burdensome and causes students get distracted in order to sign for their attendance. Furthermore, the face recognition based automated student attendance system able to overcome the problem of fraudulent approach and lecturers does not have to count the number of students several times to ensure the presence of the students.

1.2 Project Overview

The project “Face Attendance” mainly focuses on –

1. Reducing time wastage during conventional class attendance.

2. Utilizing latest trends in machine vision to implement a feasible solution for class attendance system.
3. Automating the whole process so that we have digital environment.
4. Preventing fake roll calls as one to one attendance marking is possible only.
5. Encouraging the use of technology in daily lives.

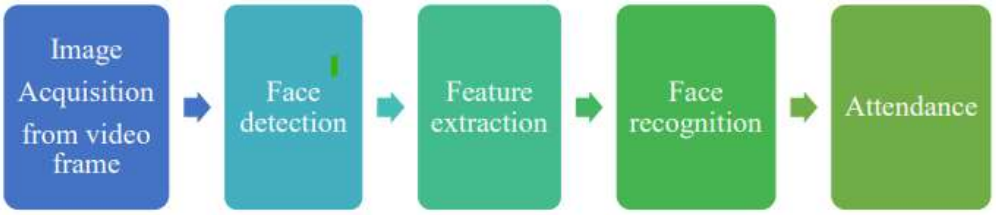


Figure 1.1 Block Diagram of the General Framework

1.3 Hardware Specification

- CPU (3.0 GHz or faster) or faster 64-bit Dual Core processor like Intel core-2 duo.
- Memory: 4GB(DDR4 | DDR2) RAM or more
- Camera for capturing the image
- Speaker (1mW) i.e. like in ear microphone.

1.4 Software Specification

- Python Interpreter
- Operating system: Linux- Ubuntu 16.04 to 17.10

CHAPTER-2

LITERARTURE SURVEY

2.1 EXISTING SYSTEM

There are some systems like – Multimedia Management, Law Enforcement, which uses cameras and other stuffs to identify the Face/objects. However, there is no such existing system to help to reduce time wastage to taking the attendance.

2.2 PROPOSED SYSTEM

In the face detection and recognition system, the process flow is initiated by being able to detect the facial features from a camera or a picture store in a memory. The algorithm processes the image captured, identifies the number of faces in the image by analyzing from the learned pattern, and compare them to filter out the rest. This image processing uses multiple algorithm that takes facial features and compare them with known database. The motivation behind this project is to simplify the means by which attendance is taken during lectures and how much time it takes. The use of ID cards or manually calling out attendance and writing it down on sheets is not productive and efficient. This system will detect the number of faces on the class and will identify them from the store database. With the face detection and recognition system in place, it will be easy to tell if a student is actually present in the classroom or not.

2.3 FEASIBILITY STUDY

- Financial Stability: The price of the equipment will not be too high and minimal cost will be charged from the people, which include the cost of the hardware and the software.

- Technical feasibility: Each of the hardware and the software used are freely available in the market and the technologies used are open source which means anyone can contribute in these technologies. The data collected from the user will be stored in the user local system and it will be used to improve the accuracy and the functioning of the application.

- Economic Feasibility: Economic feasibility defines whether the expected benefit equals or exceeds the expected costs. It is also commonly referred to as cost/benefit analysis. The procedure is to determine the benefits and the savings expected from the system and compare them with the costs. A proposed system is expected to outweigh the costs.

- Operational Feasibility: Operational feasibility is the measure of how well a proposed system solves the problems with the users. Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented. The project is operationally feasible for the users as nowadays almost all the teachers/staffs are familiar with digital technology.

CHAPTER-3

SYSTEM ANALYSIS & DESIGN

3.1 Requirement Specification

3.1.1 PYTHON

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

3.1.2 GOOGLE COLAB

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

3.1.3 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodie

3.1.4 TENSOR FLOW

Tensor flow is an open-source software library for dataflow and differentiable programming across a range of tasks. It is an symbolic math library, and is also used for machine learning application such as neural networks, etc.. Google uses it for both research and production. The Google Brain team for internal Google use develops tensor flow. It is released under the Apache License 2.0 on November 9,2015. Tensor flow is Google Brain's second-generation system.1st Version of tensor flow was released on February 11, 2017.While the reference implementation runs on single devices, Tensor flow can run on multiple CPU’s and GPU (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units). Tensor Flow is available on various platforms such as64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS. The architecture of tensor flow allows the easy deployment of computation across a variety of platforms (CPU’s, GPU’s, TPU’s), and from desktops - clusters of servers to mobile and edge devices. 22 Tensor flow computations are expressed as tasteful dataflow graphs. The name Tensor flow derives from operations that such neural networks perform on multidimensional data arrays, which are referred to as tensors.

3.1.5 NUMPY

NumPy is library of Python programming language, adding support for large, multi-dimensional array and matrice, along with large collection of high-level mathematical function to operate over these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several developers. In 2005 Travis Olphant created NumPy by incorporating features of computing Numarray into Numeric, with extension modifications. NumPy is open-source software and has many contributors.

3.1.7 VISUAL STUDIO

Visual Studio Code is a streamlined **code** editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick **code**-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as **Visual Studio** IDE.

3.2 Flowchart

3.2.1 Face Attendance Model

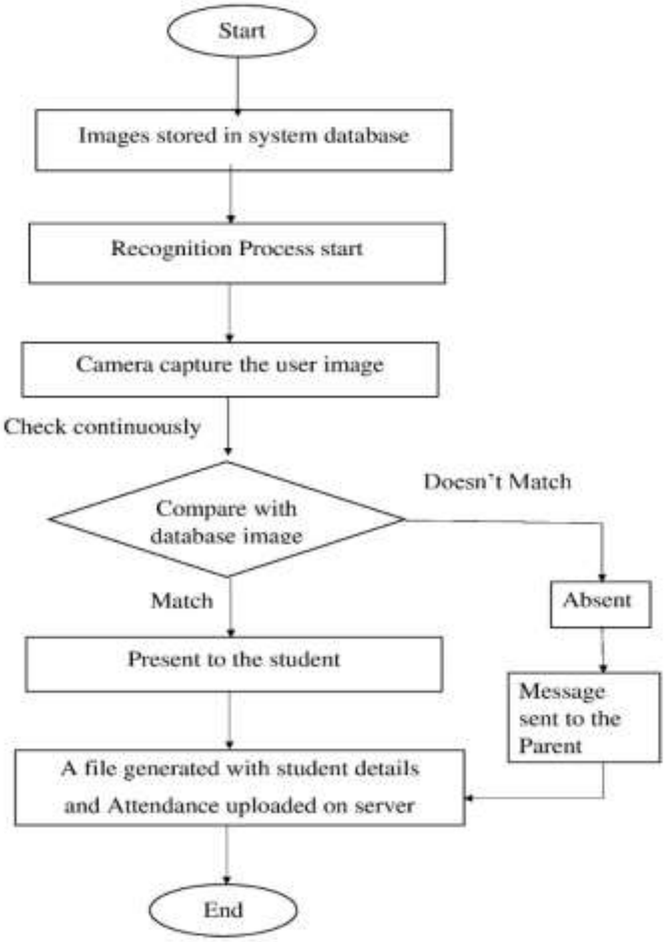

Start
Images stored in system database
Recognition Process start
Camera capture the user image
Check continuously
Compare with database image
Doesn't Match
Match
Absent
Present to the student
Message sent to the Parent
A file generated with student details and Attendance uploaded on server
End

Fig 3.2 Flowchart of Face Attendance Model

Explanation:

Face recognition extends beyond detecting the presence of a human face to determine whose face it is. The process uses a computer application that captures a digital image of an individual's face -- sometimes taken from a video frame -- and compares it to images in a database of stored records.

Face detection helps identify which parts of an image or video should be focused on to determine age, gender and emotions using facial expressions. In a facial recognition system -- which maps an individual's facial features mathematically and stores the data as a faceprint -- face detection data is required for the algorithms that discern which parts of an image or video are needed to generate a faceprint. Once identified, the new faceprint can be compared with stored faceprints to determine if there is a match.

3.2.2 Object Detection Model

Explanation:

Object detection is a computer vision technique that works to identify and locate objects within an image or video. Specifically, object detection draws bounding boxes around these detected objects, which allow us to locate where said objects are in (or how they move through) a given scene.

An encoder takes an image as input and runs it through a series of blocks and layers that learn to extract statistical features used to locate and label objects. Outputs from the encoder are then passed to a decoder, which predicts bounding boxes and labels for each object. The bounding boxes are then shown to us as the output of the object identified in the picture by the model. The model predicts the bounding boxes of the object after verifying it with the help of accuracy.

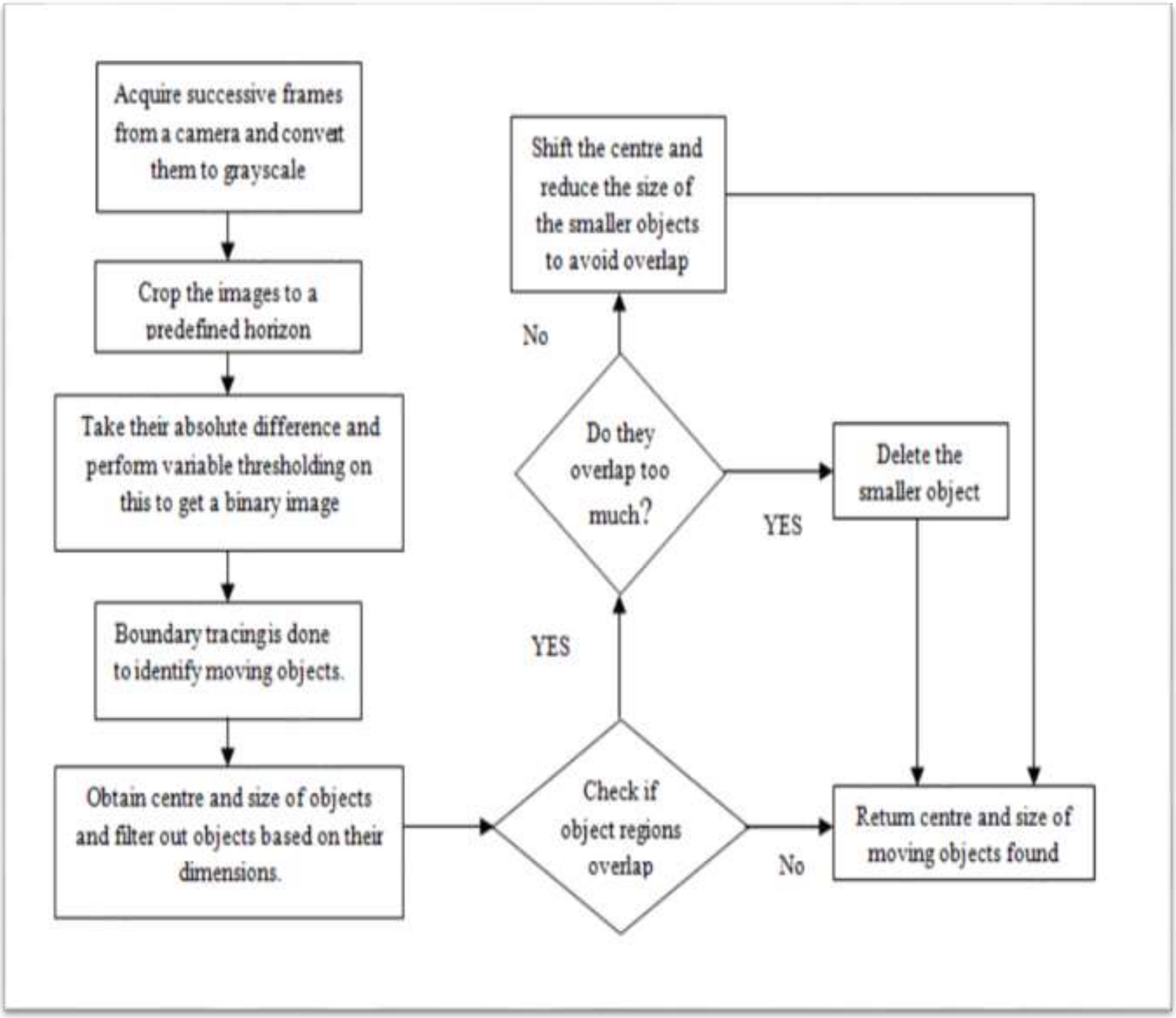

Acquire successive frames from a camera and convert them to grayscale
Crop the images to a predefined horizon
Take their absolute difference and perform variable thresholding on this to get a binary image
Boundary tracing is done to identify moving objects.
Obtain centre and size of objects and filter out objects based on their dimensions.
Check if object regions overlap
YES
No
Do they overlap too much?
No
YES
Shift the centre and reduce the size of the smaller objects to avoid overlap
Delete the smaller object
Return centre and size of moving objects found

Fig 3.3 Flowchart for Object Detection Model

3.3 Sequence Diagram

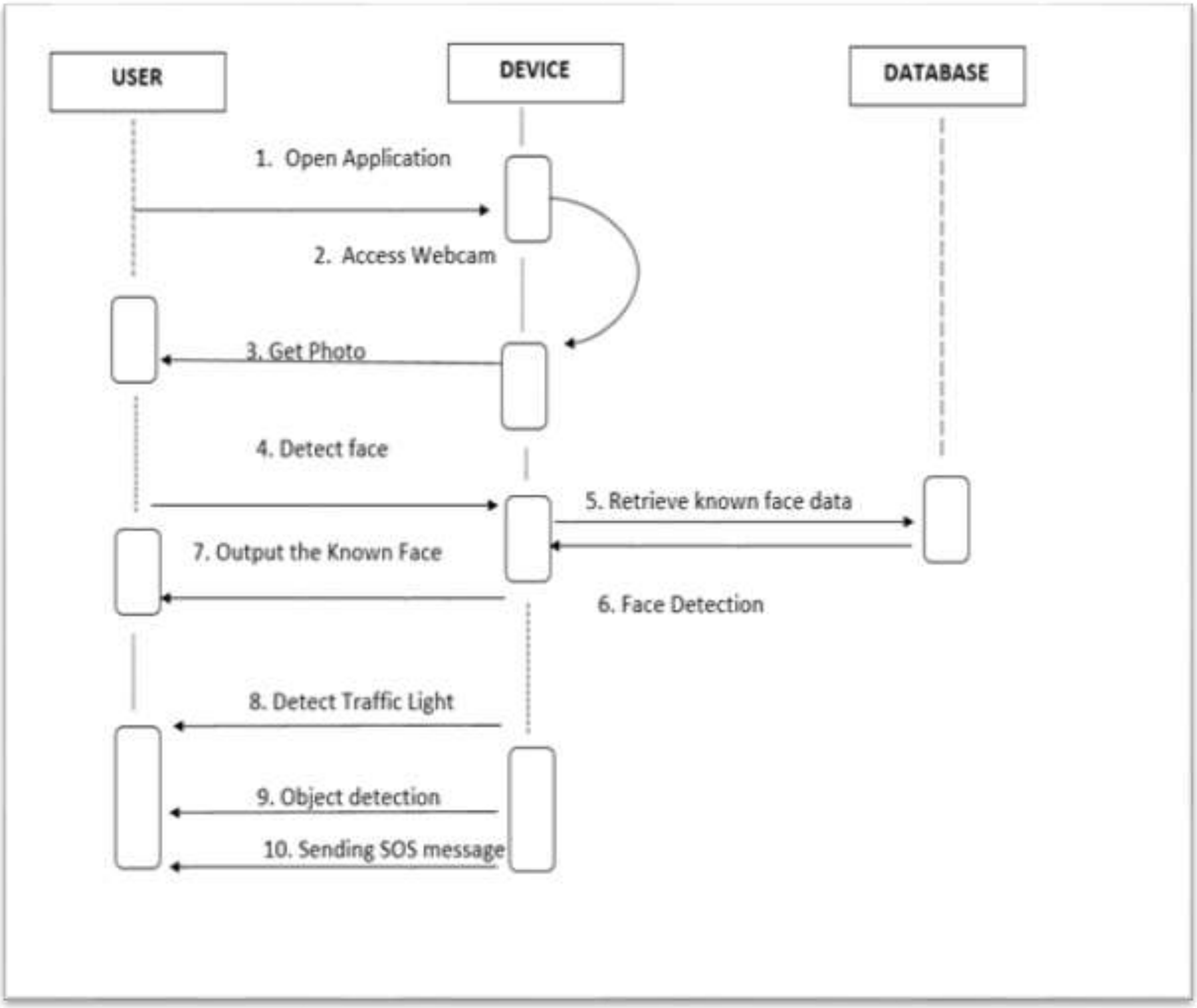


Fig 3.4 Sequence Diagram

Explanation:

The use case diagram is explained as follows:

A user opens the application and the web cam is accessed which captures the live camera feed from the environment. The image captured from the background is then transferred to the face detection and the object detection model, which looks for the familiar faces and the object in the camera feed and tells about the output in the form of voice. The model also offers traffic light detection application to the user which can easily help the user to identify the traffic light. The user can also send the sos message in case of any emergency with the help of the application.

3.4 Use case Diagram

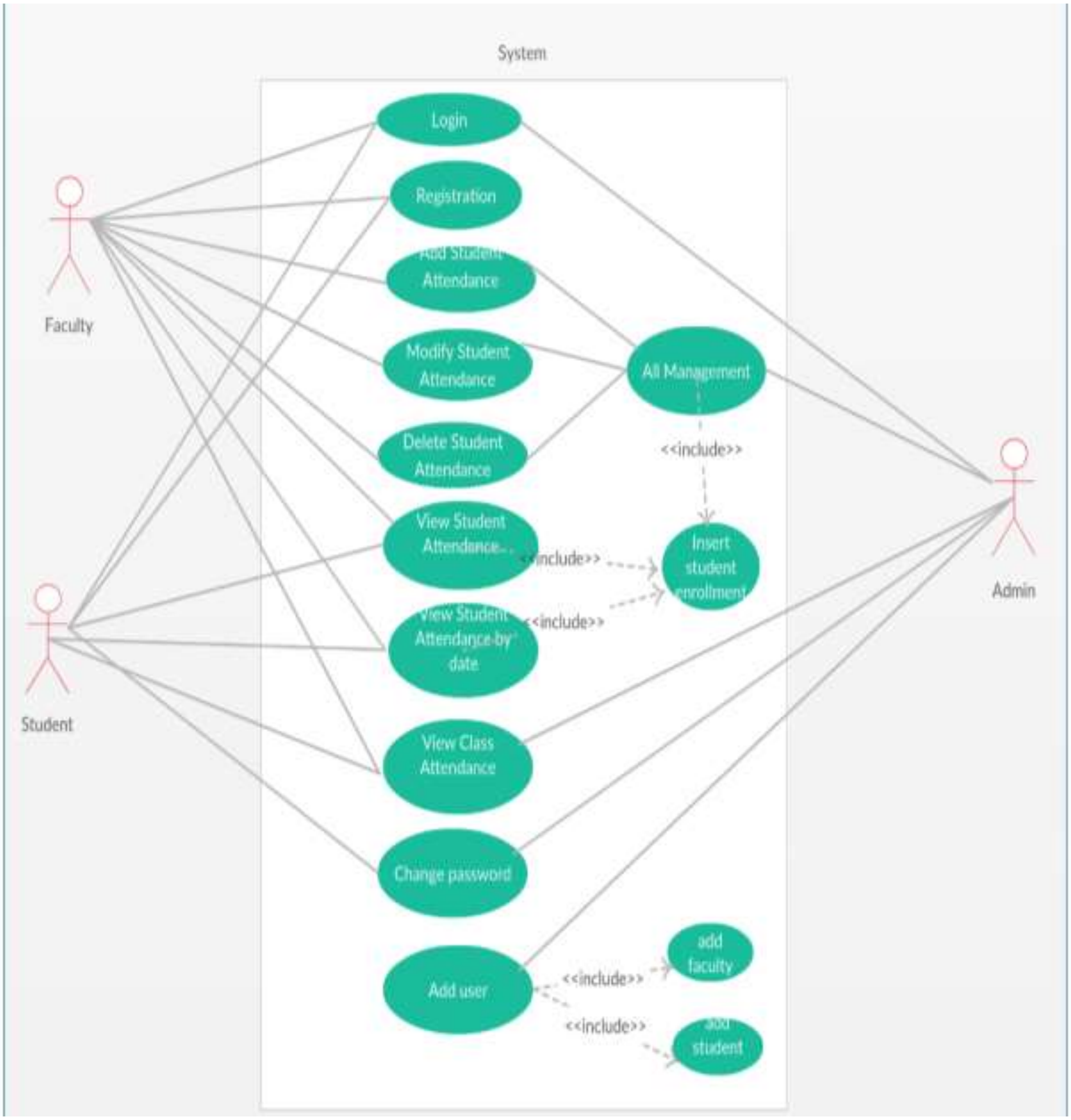


Fig 3.5 Use Case Diagram

Explanation:

The user starts the system and then the live feed from the camera is taken as input for the object detection and the face detection program. The face detection model scans the image feed and check for the known faces available in the database, which are then matched with the input data. The result is then output to the screen with the bounding boxes and the name of the person if identified is pronounced or spoken by the program. The obstacle detection program scans the image feed, detects the object if present in the image feed, and tells the output with accuracy in the form of speech.

The system also provides the facility to send an SOS signal to the users contact list in case of any emergencies.

CHAPTER-4

RESULTS/OUTPUTS

With the help of this project, students are able to reduce the wastage during conventional class attendance. Recognizing known faces and help preventing fake roll calls as one to one attendance marking is possible only, encouraging the use of technology in daily lives, utilizing latest trends in machine vision to implement a feasible solution for class attendance system.

By using this thesis and based on experimental results we are able to detect face more precisely and identify the faces individually with exact location of an faces in the picture in x, y axis.

SCREENSHOTS

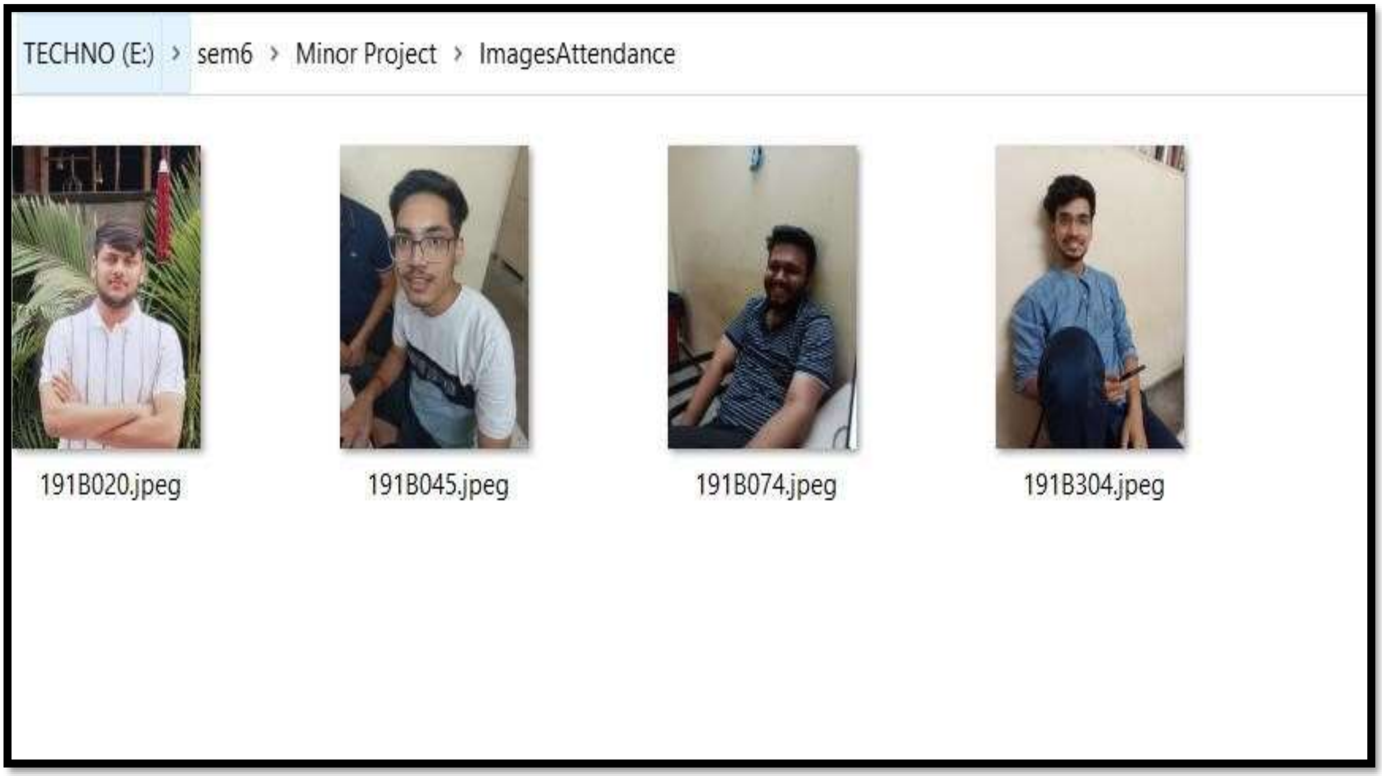


Fig 3.6

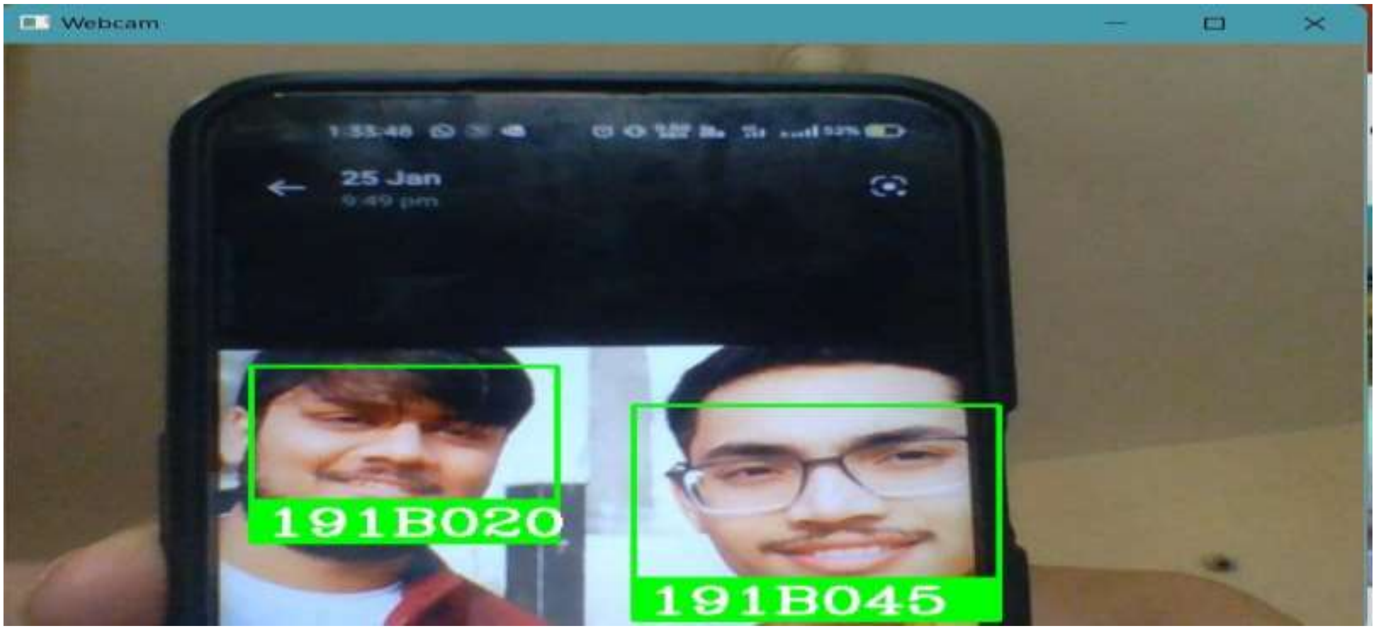
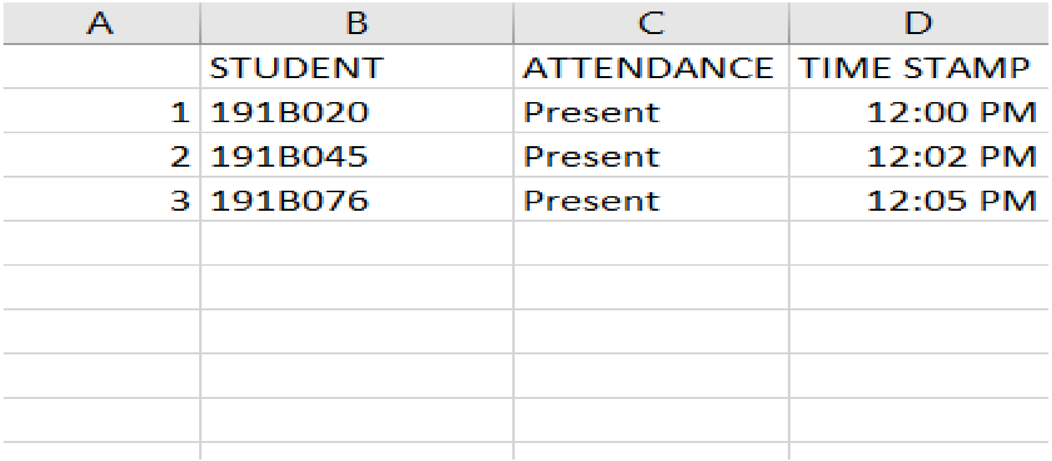


Fig 3.7 Detect more than one face



A	B	C	D
	STUDENT	ATTENDANCE	TIME STAMP
1	191B020	Present	12:00 PM
2	191B045	Present	12:02 PM
3	191B076	Present	12:05 PM

Fig 3.9 Attendance after face detection

CHAPTER-5

CONCLUSIONS/RECOMMENDATIONS

- “Face Attendance” system provides convenience in different fields like institutions, hospitals, and business.
- At the end of this project, we expect that our application is capable of giving desired output to the users and is able to help them.
- The main goal of the project is to help the Student, teachers, and make their life easy by reducing time.
- At the end of this project, we expect that this project will identify the person face and record the attendance on excel sheet.

The system can be made more flexible and scalable using these recommendations. Please note that the system implemented here is just a prototype of idea presented via this project. The recommendations are as follows:

- The system can be extended to more number of students with freedom to change list of students according to class changes.
- The system can be made more flexible to allow updating of templates in case student incurs significant amount of change in his facial features.
- The system can also be extended to allow better face recognition algorithm in which even rotational features of face can be detected efficiently.

CHAPTER-6

REFERENCES

- Learning OpenCV –Computer Vision with the openCV Library O’Reilly Publication Gary Bradsk1 and Andrian Kehlar.

- Kyungnam Kim ” Face Recognition using Principle Component Analysis” D. C. a. T. Cootes, Boosted Regression Active Shape Models, BMVC, 2007.

- Y. W. F. W. a. J. S. X. Cao, "Face alignment by Explicit Shape Regression," in 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012.

- M. T. a. A. Pentland, "Eigenfaces for recognition," Journal of Cognitive Neuroscience, pp. 71-86, 1991