Dr. Babasaheb Ambedkar Marathwada University, Aurangabad

Department of Computer Science and Information Technology

(2022-2023)

A Major Project Report On

"FACE RECOGNITION BASED ATTENDANCE SYSTEM"

Under the Guidance of

Dr.Seema S. Kawathekar
(Assistant Professor)

Submitted By

Mr. Arun Sahadev Ghodake

Miss. Punam Raghunath Kadam

Dr. Babasaheb Ambedkar Marathwada University, Aurangabad.

Department of Computer science and Information Technology

CERTIFICATE

This is to certify that the project report entitled **“FACE RECOGNATION BASED ATTENDANCE SYSTEM”** has been Submitted by **Mr. Arun S. Ghodake** and **Miss. Punam R. Kadam**, student of M.SC (CS&IT) -IIndYear (IVth SEM). Department of Computer Science and Information Technology, Aurangabad.

In the partial fulfilment for the requirement of award Master of Information Technology degree of Dr. Babasaheb Ambedkar Marathwada University, Aurangabad in the academic year 2022-2023 is a record of student own study carried under my supervision and guidance.

This report has not been submitted to any other university or institution for the award of the any degree.

Seat Number:_______

Seat Number:_______

Paper Code:

Date:

Dr.Seema S. Kawathekar
(Assistant Professor)
Guide

Examiner

Prof. Sachin N.Deshmukh
(HOD)

Acknowledgement

The success of project is an outcome of sincere efforts, channeled in right direction, efficient supervision and most valuable technical assistance. This project wouldn't have been completed without the help express by my special gratitude to, **Prof. Sachin N. Deshmukh Sir** Head of department of Computer Science and **Dr.Seema S. Kawathekar** Project Guide.

We would also like to acknowledge all the staffs for providing a helping hand to us in times of queries & problems. The Project is a result of the efforts of all the peoples who are associated with the Project directly or indirectly, who helped us to successfully complete the Project within the specified Time Frame.

Courage & Support, the Project Development would have been Futile It was only their building Support & Morale us in attaining the Successful completion of the Project.

My Senior Research Colleagues Mr. Dinesh M. Barode and Miss. Rupali S. Awhad Also We would like to Thanks our colleagues for keeping our Sprits High while preparing the Project. Because of their Diligent & Hard Work, we wouldn't have been able to complete the Project within the given Time Frame.

We are Thankful to each & every people involved with us in this case study, their Encouragement & Support enabled the Project to Materialize & Contributed it to its success. We would like to express our Appreciation to all the people who have contributed to the Successful completion of the Project.

With all Respects & Gratitude, we would like to Thanks to all the people, who have helped in the Development of the Project

Thank You Very much....!

Mr. Arun Sahadev Ghodake

Miss. Punam Raghunath Kadam

Abstract

The system is developed for deploying an easy and a secure way of taking down attendance. The software first captures an image of all the authorized persons and stores the information into database. The system then stores the image by mapping it into a face coordinate structure.

The main goal of this project is to create a Face Recognition-based attendance system that will turn this manual process into an automated one. This project meets the requirements for bringing modernization to the way attendance is handled, as well as the criteria for time management.

The image is processed as follows: first, faces are identified using a Haar cascade classifier, then faces are recognized using the LBPH (Local Binary Pattern Histogram) Algorithm, histogram data is checked against an established dataset, and the device automatically labels attendance.

Index

LIST OF FIGURES

LIST OF TABLES

1. Introduction

This is a project for educational institutions on a facial recognition-based attendance system. The traditional way of marking commute to work can be a tedious task in many skills and colleges. It is also an additional responsibility for schools to mark an attendance by naming Individuals which may take up to 5 minutes for a full session. This can be time consuming. There are certain chances of a representative. Therefore, many institutions began using many other recording methods such as the use of Frequency Identification (RFID), Iris recognition, fingerprint recognition, and more. However, these online based systems can be time consuming and disruptive in nature. Face recognition has set a very important biometric feature, which is easily accessible and unobtrusive. Face-based systems that ignore a lot of facial expression. The face recognition program consists of two stages:

1. Face verification

2. Facial recognition.

Face verification is a 1: 1 matching process, comparing face-to-face image processing and there is a 1: N comparing face query image. The purpose of this approach is to create a travel plan based on face recognition strategies. Here a personal face is considered a mark of presence. Today, facial recognition is gaining popularity and has been widely used. In the middle of this page, we suggested a face recognition system in the live video class and attendees will be marked when they found face is found within the website. In comparison to existing procedures, this new technology will take less time.

The system is developed for deploying an easy and a secure way of taking down attendance. The software first captures an image of all the authorized persons and stores the information into database. The system then stores the image by mapping it into a face coordinate structure.

A facial recognition attendance system incorporates facial recognition technology to recognize and verify an employee's facial features and to record attendance automatically.

1.1 Biometrics

Biometric face recognition attendance systems utilize facial recognition technology to accurately identify individuals and record their attendance in various settings, such as workplaces, educational institutions, or events. These systems rely on unique facial features and patterns to verify the identity of individuals.

Here's a general overview of how a biometric face recognition attendance system works:

Enrolment: Initially, individuals need to be enrolled in the system. During enrolment, their facial biometric data is captured and stored in a database. This typically involves taking a series of images or video of the person's face from different angles to create a comprehensive representation of their facial features.

Face Detection: When attendance is to be recorded, the system uses cameras or other imaging devices to capture a live image or video stream. It employs face detection algorithms to locate and isolate faces within the captured data.

Face Recognition: Once a face is detected, the system applies face recognition algorithms to compare the facial features extracted from the captured image or video with the facial templates stored in the database during enrolment. These algorithms analyze various facial attributes such as the distance between the eyes, shape of the nose, and contour of the face to determine a match.

Attendance Recording: If a match is found between the captured image and the enrolled templates, the system records the attendance of the individual associated with the recognized face. The attendance data can be stored locally or sent to a centralized database for further processing and analysis.

Verification and Accuracy: Biometric face recognition systems often employ techniques to enhance accuracy and security. These may include liveness detection mechanisms to ensure that the captured data is from a live person and not a photograph or video playback. Additionally, the system can incorporate machine learning algorithms to continuously improve its recognition performance over time.

Biometric face recognition attendance systems offer several advantages, including improved efficiency, reduced administrative burden, and enhanced accuracy compared to traditional attendance methods. They eliminate the need for physical identification cards or passwords and provide a reliable and convenient way to record attendance based on unique facial characteristics.

1.2 Project Objective

A face recognition attendance system automatically identifies and confirms a person and records attendance based on their face detection. Face recognition attendance systems are catching the attention of both small and large businesses.

Attendance is prime important for both the teacher and student of an educational organization. So it is very important to keep record of the attendance. The problem arises when we think about the traditional process of taking attendance in class room.

The main purpose of this project is to build human face recognition for an institute or organization to mark the attendance of their students or employees. It is a sub domain of Object Detection, where we try to observe the instance of semantic objects. This system is fully automated and easily deployable.

1.3 Background

Face recognition is crucial in daily life in order to identify family, friends or someone we are familiar with. We might not perceive that several steps have actually taken in order to identify human faces. Human intelligence allows us to receive information and interpret the information in the recognition process. We receive information through the image projected into our eyes, by specifically retina in the form of light. Light is a form of electromagnetic waves which are radiated from a source onto an object and projected to human vision

1.4 Overview of the Project

A facial recognition attendance system uses facial recognition technology to identify and verify a person using the person's facial features and automatically mark attendance. The software can be used for different groups of people such as employees, students, etc. The system records and stores the data in real-time.

The image is processed as follows: first, faces are identified using a Haarcascade classifier, then faces are recognized using the LBPH (Local Binary Pattern Histogram) Algorithm, histogram data is checked against an established dataset, and the device automatically labels attendance.

1.5 Problem Statement

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 5 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 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 do not have to count the number of students several times to ensure the presence of the students.

Hence, there is a need to develop a real time operating student attendance system which means the identification process must be done within defined time constraints to prevent omission. The extracted features from facial images which represent the identity of the students have to be consistent towards a change in background, illumination, pose and expression. High accuracy and fast computation time will be 6 the evaluation points of the performance.

2. Literature Review

Dr. V Suresh, Srinivasa ChakravarthiDumpa, Chiranjeevi Deepak Vankayala, HaneeshaAduri, Jayasree Rapa, “Facial Recognition Attendance System Using Python and Opecv”, The main purpose of this project is to build a face recognition-based attendance monitoring system for educational institution to enhance and upgrade the current attendance system into more efficient and effective as compared to before. The current old system has a lot of ambiguity that caused inaccurate and inefficient of attendance taking. Many problems arise when the authority is unable to enforce the regulation that exists in the old system. The technology working behind will be the face recognition system. The human face is one of the natural traits that can uniquely identify an individual. Therefore, it is used to trace identity as the possibilities for a face to deviate or being duplicated is low.

In this project, face databases will be created to pump data into the recognizer algorithm. Then, during the attendance taking session, faces will be compared against the database to seek for identity. When an individual is identified, its attendance will be taken down automatically saving necessary information into a excel sheet. At the end of the day, the excel sheet containing attendance information regarding all individuals are mailed to the respective faculty [1].

Yohei KAWAGUCHI Tetsuo SHOJI WeijaneLIN Koh KAKUSHO Michihiko MINOH “Face Recognition-based Lecture Attendance System”, In this paper, we propose a system that takes the attendance of students for classroom lecture. Our system takes the attendance automatically using face recognition. However, it is difficult to estimate the attendance precisely using each result of face recognition independently because the face detection rate is not sufficiently high. In this paper, we propose a method for estimating the attendance precisely using all the results of face recognition obtained by continuous observation. Continuous observation improves the performance for the estimation of the attendance. We constructed the lecture attendance system based on face recognition, and applied the system to classroom lecture. This paper first review the related works in the field of attendance management and face recognition. Then, it introduces our system structure and plan. Finally, experiments are implemented to provide as evidence to support our plan. The result shows that continuous observation improved the performance for the estimation of the attendance [2].

Seema S. Kawathekar[1], Dinesh M. Barode[2], Rupali S. Awhad[3], Arun S. Ghodake[4] "Automated Attendance system using face detection: A Hybrid approach**",**
Automated attendance system is more useful for educational institution for upgrading and enhancing their conventional attendance system. The convectional system very time consuming. It has lot of ambiguity. Instead of this hear we include various biometric techniques like thumb, palm, voice, face. Our paper face recognition system is use to identified and individual student. Face Recognition is a computer programme that can locate, recognize, or confirm students' faces in a image or video that was taken with a digital camera .This is very useful in taking attendance in school, colleges, offices, etc. Face detection includes several categories they are feature-based, appearance-based, knowledge-based, and template matching. We looked into many methods and concluded that the Haar cascade classifier is among the most effective for detecting the object. An algorithm for detecting objects called Haar cascade produces findings that are accurate results. In this study, Haar cascade classifier is used, in the Real-time face detection gives RGB (Red, Green, Blue) colour image or video are converted into grayscale, it provides us with a single value for the image, which is then compared with our datasets to find faces .When a student is identified its attendance we take automatically into .csv file and the report is generated by date, moths or year for necessary actions. Like exam form filling, hall-ticket generation etc [3].

Hafiza MahrukhShahzadi , and Ms. Shazia Riaz**"**Faculty Members Presence in Department by Face Recognition (FMPFR)," We have proposed a method for real time human detection & counting of visitors in hotel using a web camera. Normally when an employee get a break of one or two hour for launch. He went to a hotel for launch but hotel is fully reserved. He went to another hotel but situation is same there. That is also fully reserved. In this way he lost his maximum time of launch break. This is a big issue that mostly people lost their break time in finding place in hotel. In this approach we will design a website where list of hotel of a different cities are present and when the user select his desired hotel name he will see that how many place in that hotel is available and how many tables are reserved in hotel. This enlisting is a onetime procedure and their face will be put away in the database. Amid enlisting of face we require a system since it is a onetime procedure. You can have your own move number as your representative id which will be novel for every worker. The nearness of every worker will be refreshed in a database. The outcomes indicated enhanced execution over manual participation administration system this item gives significantly more arrangements with exact outcomes in client intuitive way instead of existing leave administration system[4].

Naveed Khan Balcoh, M. Haroon Yousaf, Waqar Ahmad and M. Iram Baig,“Algorithm for Efficient Attendance Management: Face Recognition based approach”, This paper introduces the efficient and accurate method of attendance in the classroom environment that can replace the old manual methods. This method is secure enough, reliable and available for use. No need for specialized hardware for installing the system in the classroom. It can be constructed using a camera and computer. There is a need to use some algorithms that can recognize the faces in veil to improve the system performance [5].

Ms. Sarika Ashok Sovitkar[l], Dr. Seema S. Kawathekar[2,] “A Conceptual Model for Automated Attendance System Using Principal Component Analysis (PCA),”In this paper Traditional method of attendance marking is very time consuming and becomes complicated when the strength is more. Automated Attendance System is the advancement that has taken place in the field of automation replacing traditional attendance marking activity. Automated Attendance Systems are generally bio-metric based, smart-card based and web based' These systems are widely used in different organizations. Automation of Attendance System has edge over traditional method as it saves time and also can be used for security purposes. This also helps to prevent fake attendance, An Attendance Management System which is developed using biometrics, in our case face, generally consists of Image Acquisition, Database development, Face detection (Viola-Jones), Pre-processing, Feature extraction (Eigen Faces), and Classification stages followed by Post processing stage. Face recognition is also a very promising application of biometric technology bec4use it does not need physical contact with any device [6].

PAUL VIOLA Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA MICHAEL J. JONES Mitsubishi Electric Research Laboratory, 201 Broadway, Cambridge, MA 02139, USA “Robust Real-Time Face Detection,” This paper brings together new algorithms and insights to construct a framework for robust and extremely rapid visual detection. Toward this end we have constructed a frontal face detection system which achieves detection and false positive rates which are equivalent to the best published results (Sung and Poggio, 1998; Rowley et al., 1998; Osuna et al., 1997a; Schneiderman and Kanade, 2000; Roth et al., 2000). This face detection system is most clearly distinguished from previous approaches in its ability to detect faces extremely rapidly. Operating on 384 by 288 pixel images, faces are detected at 15 frames per second on a conventional 700 MHz Intel Pentium III. In other face detection systems, auxiliary information, such as image differences in video sequences, or pixel color in color images, have been used to achieve high frame rates. Our system achieves high frame rates working only with the information present in a single grey scale image. These alternative sources of information can also be integrated with our system to achieve even higher frame rates[7].

Anurag Pandey, Divyansh Choudhary, Ritik Agarwal, Tushar Shrivastava, Kriti*“Face detection using Haar cascade classifier,”In the present work the authors have proposed the use of Haar cascade classifier to detect the frontal face and left profile of a person at a time. Till now in the present work, the authors have been able to detect only the frontal profile (frontal face) and left profile of a human face. The work can further be extended to include the right profile detection of the person as well as detecting multiple faces at a time. The developed system can be used for detecting faces for various purposes such as catching criminals, emotion detectors, law enforcement, finding a missing person, marketing and advertising, airport and border control or even as an attendance marking tool for marking attendance in schools, colleges, institutions, workplaces etc. An advanced version of this like eye blink detection can be used to warn people if they are sleepy while driving and that will reduce the number of road accident [8].

3.Modal Implementation

The main components used in the implementation approach are open source computer vision library (OpenCV). One of OpenCV's goals is to provide a simpleto - use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. OpenCV library contains over 500 functions that span many areas in vision. The primary technology behind Face recognition is OpenCV.

Haar Cascade is a feature-based object detection algorithm to detect objects from images. A cascade function is trained on lots of positive and negative images for detection. The algorithm does not require extensive computation and can run in real-time.

3.1 Design Requirements

We used some tools to build the system. Without the help of these tools it would not be possible to make it done. Here we will discuss about the most important one.

3.2 Software Implementation

1. Python: python is an interpreted, high-level, general purpose programming language created by Guido Van Rossum and first released in 1991, Python's design philosophy emphasizes code Readability with its notable use of significant Whitespace. Its language constructs and 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 procedural, objectoriented, and functional programming.

2. OpenCV: We used OpenCV 3 dependency for python 3. OpenCV is library where there are lots of image processing functions are available. This is very useful library for image processing. Even one can get expected outcome without writing a single code. The library is cross-platform and free for use under the open-source BSD license. Example of some supported functions are given bellow:

- **Histograms:** computing, equalization, and object localization with back projection algorithm
- **Segmentation:** thresholding, distance transform, foreground/background detection, watershed segmentation
- **Filtering:** linear and nonlinear filters, morphological operations
- **Cascade detectors:** detection of face, eye, car plates
- **Video processing:** optical flow, background subtraction, camshaft (object tracking)
- **Photography:** panoramas realization, high definition imaging (HDR), image inpainting

3. PIP: It is the package management system used to install and manage software packages written in Python.

4. NumPy: NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object.
- Sophisticated (broadcasting) functions

3.3 Hardware Implementation

The section of hardware configuration is an important task related to the software development. Insufficient random access memory may affect adversely on the speed and efficiency of the entire system. The process should be powerful to handle the entire operations. The hard disk should have sufficient capacity to store the file and application.

1. 4 GB RAM
2. 80 GB HDD
3. Dual Core processor
4. CDROM (installation only). VGA resolution monitor
5. Microsoft Windows 98/2000/NT with service pack 6 / XP with service pack 2

4. Methodology

The proposed system is designed for automating the attendance of the different organization and reduces the flaws of existing manual system. The system calculate the attendance subject wise, that is the data of students and subjects are added manually by administrator, and whenever time for corresponding subject arrives the system automatically starts taking snaps and find whether human faces are appear in the given image or not. We have used Histogram of Oriented Gradient for face detection and deep learning techniques to calculate and compare 128-d face features for face recognition. Once faces are detected and recognize with the existing database, system calculate attendance for the recognize students with the respective subject id in real time. And an excel sheet generated and saved by the system automatically. Our system splits into two parts, First the front end side which consist of GUI which is based on Electron JS that is python stack which is serving as a client and the second is the backend side which consist of logic and based on Python which is serving as a server.

4.1 Digital Image Processing

Digital image processing refers to the manipulation, enhancement, analysis, and interpretation of digital images using computer algorithms and techniques. It involves the application of various mathematical and computational operations to digital images in order to extract useful information or improve their visual quality.

Digital image processing techniques can be broadly categorized into two main areas: image enhancement and image analysis. Image enhancement techniques are used to improve the visual appearance of images by adjusting their brightness, contrast, color balance, and sharpness. These techniques can also involve noise reduction, image restoration, and image fusion.

Image analysis techniques, on the other hand, focus on extracting meaningful information from digital images. This can include object detection, image segmentation, pattern recognition, feature extraction, and image classification. These techniques are commonly used in fields such as medical imaging, remote sensing, computer vision, and scientific research.

Digital image processing relies on algorithms that operate on the pixel values of an image, allowing for the manipulation and analysis of individual or groups of pixels. These algorithms can range from simple operations, such as filtering and thresholding, to more complex techniques, such as image registration and image compression.

Overall, digital image processing plays a crucial role in a wide range of applications, including photography, multimedia, medical diagnostics, satellite imagery, surveillance, and many others. It enables the extraction of valuable information from images, enhances their visual quality, and aids in the interpretation and understanding of visual data.

4.1.1 Image Representation in a Digital Computer

An image is a 2-Dimensional light intensity function $\mathbf{f}$ ($\mathbf{x}$,$\mathbf{y}$) = $\mathbf{r}$ ($\mathbf{x}$,$\mathbf{y}$) × $\mathbf{i}$ ($\mathbf{x}$,$\mathbf{y}$) -(2.0) Where, r (x, y) is the reflectivity of the surface of the corresponding image point. i (x,y) Represents the intensity of the incident light. A digital image f(x, y) is discretized both in spatial co-ordinates by grids and in brightness by quantization. Effectively, the image can be represented as a matrix whose row, column indices specify a point in the image and the element value identifies gray level value at that point. These elements are referred to as pixels or pels. Typically following image processing applications, the image size which is used is$\mathbf{256} \times \mathbf{256}$, elements, $\mathbf{640} \times \mathbf{480}$ pels or $\mathbf{1024} \times \mathbf{1024}$ pixels. Quantization of these matrix pixels is done at 8 bits for black and white images and 24 bits for colored images (because of the three color planes Red, Green and Blueeach at 8 bits).

4.1.2 Steps in Digital Image Processing

The steps involved in digital image processing can vary depending on the specific application and goals, but here is a general overview of the common steps:

Image Acquisition: Capture or obtain the digital image using devices such as cameras, scanners, or sensors.

Image Preprocessing:

- Image resizing: Adjust the size of the image if needed.
- Color space conversion: Convert the image to a different color space if required (e.g., RGB to grayscale or HSV).
- Noise removal: Apply filters or techniques to reduce noise and artifacts in the image.
- Image enhancement: Improve the quality of the image by adjusting contrast, brightness, or sharpness.

Image Segmentation:

- Thresholding: Divide the image into regions based on pixel intensity values.
- Edge detection: Identify and highlight edges or boundaries between different regions or objects in the image.
- Clustering: Group pixels with similar properties to form distinct regions.

Feature Extraction:

- Shape analysis: Extract geometric features such as edges, corners, or shapes.
- Texture analysis: Identify and quantify patterns or textures in the image.
- Color analysis: Extract color-related features or histograms.

Image Representation and Description:

- Representation: Represent the image and its features in a suitable format for further processing or analysis.
- Description: Create a concise and meaningful representation of the image features using descriptors or feature vectors.

Image Enhancement and Restoration:

- Contrast enhancement: Adjust the contrast of the image to improve its visual quality or emphasize certain details.
- Image restoration: Recover or improve the quality of the image by removing blurring, noise, or other degradations.

Image Compression:

- Reduce the size of the image file while preserving important information.
- Apply compression algorithms such as JPEG, PNG, or others to minimize storage or transmission requirements.

Object Recognition and Classification:

- Object detection: Identify and locate specific objects or regions of interest in the image.
- Object classification: Assign labels or categories to the detected objects using machine learning or pattern recognition techniques.

Image Analysis and Interpretation:

- Extract quantitative measurements or statistical information from the image.
- Perform data analysis, visualization, or modeling based on the extracted image data.

Decision Making or Action:

Make decisions or take actions based on the analysis and interpretation of the processed image data, such as medical diagnosis, automated systems, or image-based control.

These steps are not always performed in a linear manner and can be iterative, with adjustments made at different stages based on the desired outcomes. Additionally, different techniques and algorithms can be applied at each step, depending on the specific requirements and objectives of the image processing task.

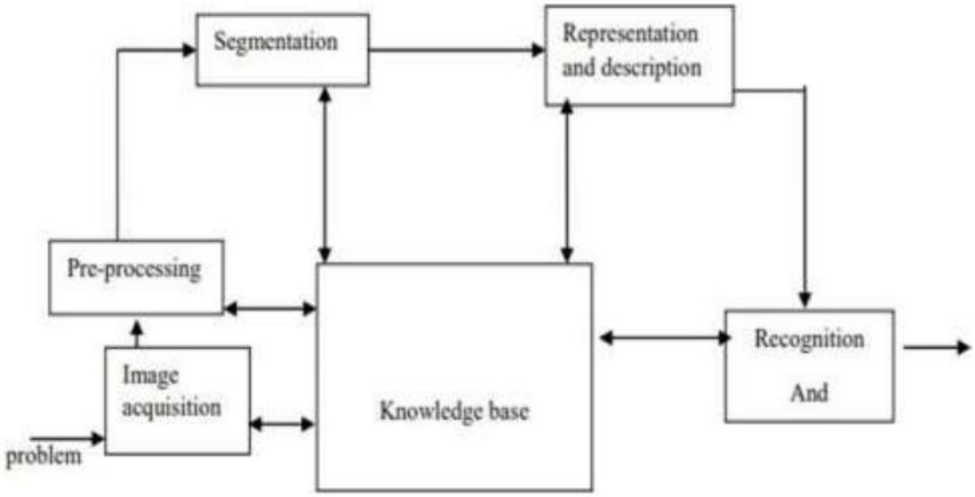


Figure 1. A diagram showing the steps in digital image processing

4.2 Data Acquisition

4.2.1 Image acquisition: Image is acquire using a high definition camera which is placed in the classroom. This image is given as an input to the system.

4.2.2 Dataset Creation: Dataset of students is created before the recognition process. Dataset was created only to train this system. We have created a dataset of 5 students which involves their name, roll number, department and images of student in different poses and variations. For better accuracy minimum 15 images of each students should be captured. Whenever we register student's data and images in our system to create dataset, deep learning applies to each face to compute 128-d facial features and store in student face data file to recall that face in recognition process. This process is applies to each image taken during registration.

4.3 Face Detection

The problem of face recognition is all about face detection. This is a fact that seems quite bizarre to new researchers in this area. However, before face recognition is possible, one must be able to reliably find a face and its landmarks. This is essentially a segmentation problem and in practical systems, most of the effort goes into solving this task. In fact the actual recognition based on features extracted from these facial landmarks is only a minor last step. There are two types of face detection problems:

1) Face detection in images

Face Detection in Images

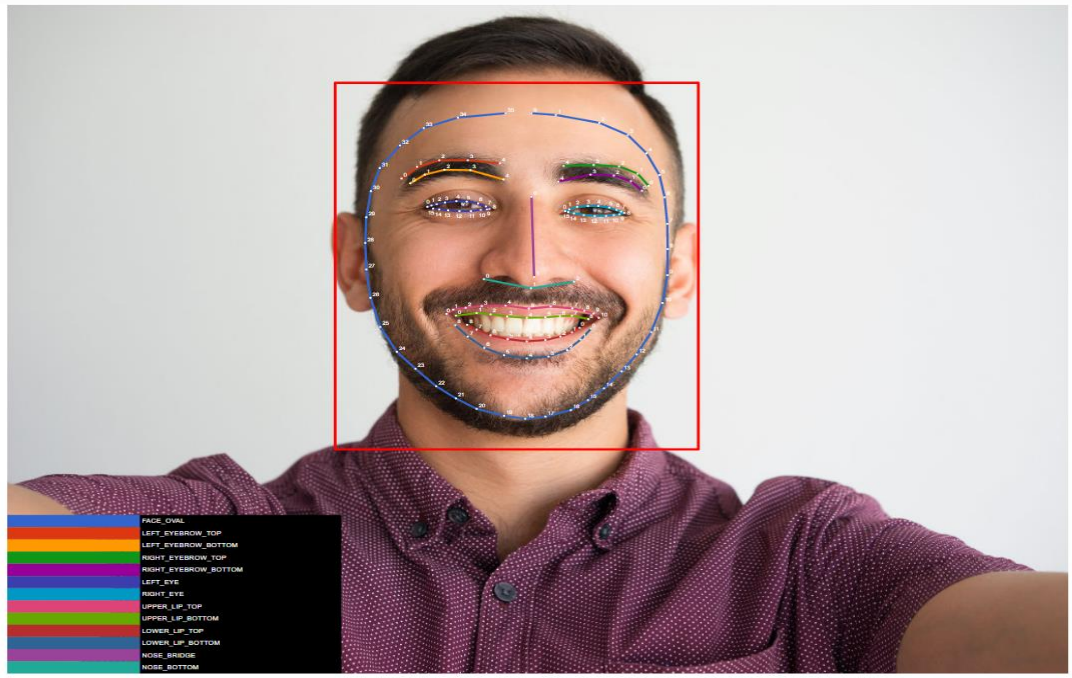
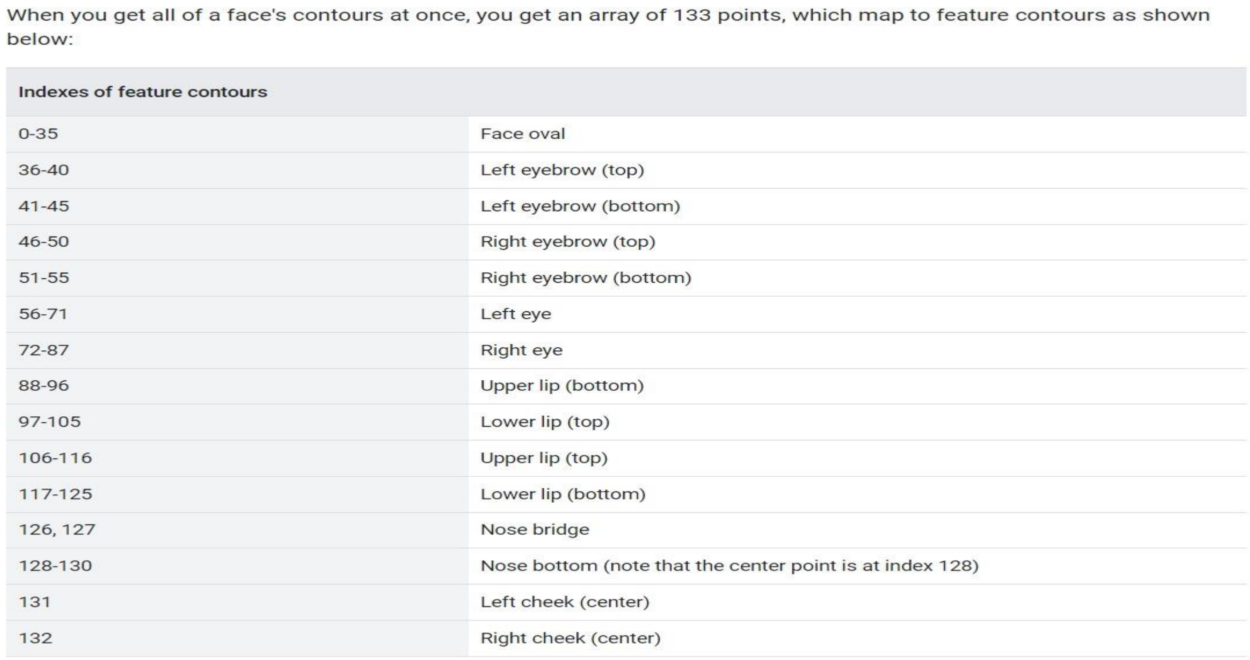


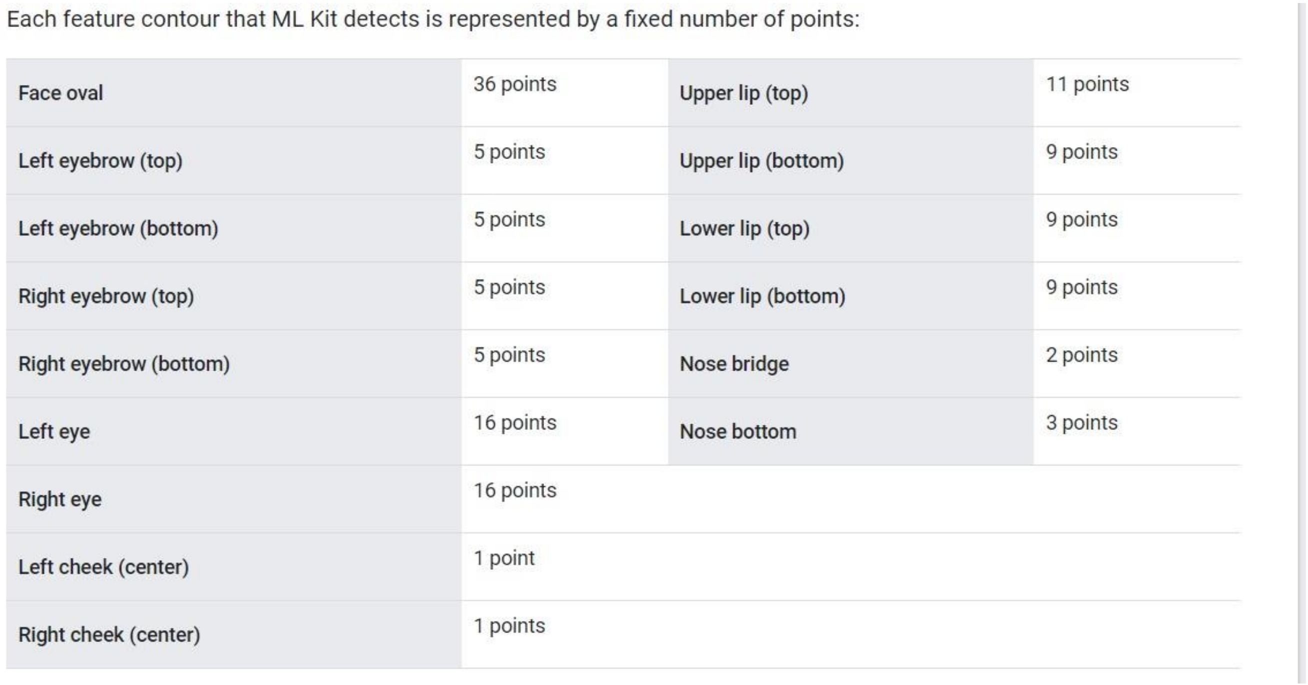
Figure 2. A successful face detection in an image with a frontal view of a human face.



When you get all of a face's contours at once, you get an array of 133 points, which map to feature contours as shown below:

Indexes of feature contours	
0-35	Face oval
36-40	Left eyebrow (top)
41-45	Left eyebrow (bottom)
46-50	Right eyebrow (top)
51-55	Right eyebrow (bottom)
56-71	Left eye
72-87	Right eye
88-96	Upper lip (bottom)
97-105	Lower lip (top)
106-116	Upper lip (top)
117-125	Lower lip (bottom)
126, 127	Nose bridge
128-130	Nose bottom (note that the center point is at index 128)
131	Left cheek (center)
132	Right cheek (center)

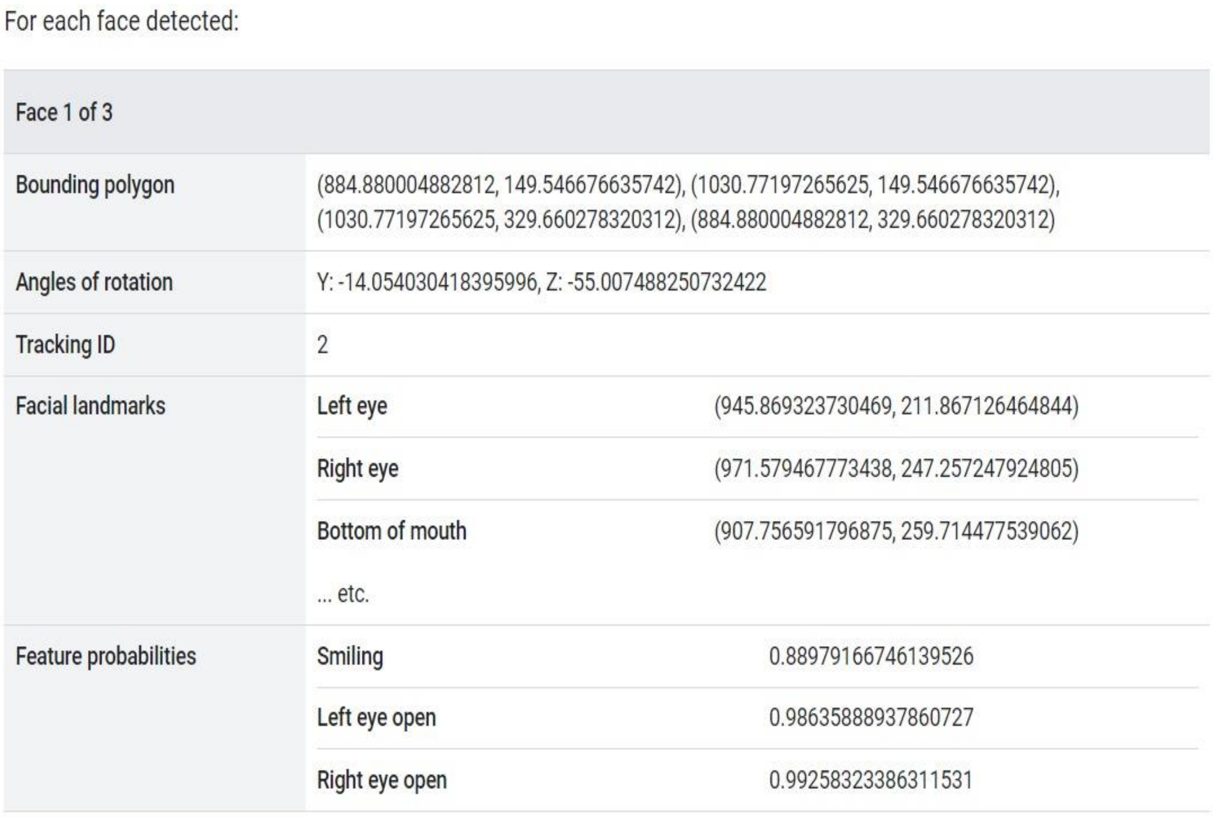
Table 1.1 Face contours 133 Points



Each feature contour that ML Kit detects is represented by a fixed number of points:

Face oval	36 points	Upper lip (top)	11 points
Left eyebrow (top)	5 points	Upper lip (bottom)	9 points
Left eyebrow (bottom)	5 points	Lower lip (top)	9 points
Right eyebrow (top)	5 points	Lower lip (bottom)	9 points
Right eyebrow (bottom)	5 points	Nose bridge	2 points
Left eye	16 points	Nose bottom	3 points
Right eye	16 points		
Left cheek (center)	1 point		
Right cheek (center)	1 points		

Table 1.2 Face contour ML kit



For each face detected:

Face 1 of 3		
Bounding polygon	(884.880004882812, 149.546676635742), (1030.77197265625, 149.546676635742), (1030.77197265625, 329.660278320312), (884.880004882812, 329.660278320312)	
Angles of rotation	Y: -14.054030418395996, Z: -55.007488250732422	
Tracking ID	2	
Facial landmarks	Left eye	(945.869323730469, 211.867126464844)
	Right eye	(971.579467773438, 247.257247924805)
	Bottom of mouth	(907.756591796875, 259.714477539062)
	... etc.	
Feature probabilities	Smiling	0.88979166746139526
	Left eye open	0.98635888937860727
	Right eye open	0.99258323386311531

Table 1.3 Face detection

4.3.1 Classification:

Classification determines whether a certain facial characteristic is present. ML Kit currently supports two classifications: eyes open and smiling. Classification is a certainty value. It indicates the confidence that a facial characteristic is present. For example, a value of 0.7 or more for the smiling classification indicates that it's likely that a person is smiling. Both of these classifications rely upon landmark detection. Also note that the classifications "eyes open" and "smiling" only work for frontal faces, i.e., faces with a small Euler Y angle (between -18 and 18 degrees).

Minimum Face Size:

The minimum face size is the desired face size, expressed as the ratio of the width of the head to the width of the image. For example, the value of 0.1 means that the smallest face to search for is roughly 10% of the width of the image being searched. The minimum face size is a performance vs. accuracy trade-off: setting the minimum size smaller lets the detector find smaller faces but detection will take longer; setting it larger might exclude smaller faces but will run faster.

The minimum face size is not a hard limit; the detector may find faces slightly smaller than specified.

Most face detection systems attempt to extract a fraction of the whole face, thereby eliminating most of the background and other areas of an individual's head such as hair that are not necessary for the face recognition task. With static images, this is often done by running a across the image. The face detection system then judges if a face is present inside the window (Brunelli and Poggio, 1993). Unfortunately, with static images there is a very large search space of possible locations of a face in an image Department of ECE Most face detection systems use an example based learning approach to decide whether or not a face is present in the window at that given instant (Sung and Poggio,1994 and Sung,1995). A neural network or some other classifier is trained using supervised learning with 'face' and 'nonface' examples, thereby enabling it to classify an image (window in face detection system) as a 'face' or 'non-face'.. Unfortunately, while it is relatively easy to find face examples, how would one find a representative sample of images which represent non-faces (Rowley et al., 1996)? Therefore, face detection systems using example based learning need thousands of 'face' and 'nonface' images for effective training. Rowley, Baluja, and Kanade (Rowley et al.,1996) used 1025 face images and 8000 non-face images (generated from 146,212,178 sub-images) for their training set! There is another technique for determining whether there is a face inside the face detection system's window - using Template Matching. The difference between a fixed target pattern (face) and the window is computed and thresholded. If the window contains a pattern which is close to the target pattern (face) then the window is judged as containing a face. An implementation of template matching called Correlation Templates uses a whole bank of fixed sized templates to detect facial features in an image (Bichsel, 1991 & Brunelli and Poggio, 1993). By using several tem

4.4 Face Detection Process

Plates of different (fixed) sizes, faces of different scales (sizes) are detected. The other implementation of template matching is using a deformable template (Yuille, 1992). Instead of using several fixed size templates, we use a deformable template (which is non-rigid) and there by change the size of the template hoping to detect a face in an image. A face detection scheme that is related to template matching is image invariants. Here the fact that the local ordinal structure of brightness distribution of a face remains largely unchanged under different illumination conditions (Sinha, 1994) is used to construct a spatial template of the face which closely corresponds to facial features. In other words, the average grey-scale intensities in human faces are used as a basis for face detection. For example, almost always an individual’s eye region is darker than his forehead or nose. Therefore an image will match the template if it satisfies the 'darker than' and 'brighter than' relationships (Sung and Poggio, 1994).

Face detection. It is process of identifying different parts of human faces like eyes, nose, mouth, etc… this process can be achieved by using MATLAB code In this project the author will attempt to detect faces in still images by using image invariants. To do this it would be useful to study the greyscale intensity distribution of an average human face. The following 'average human face' was constructed from a sample of 30 frontal view human faces, of which 12 were from females and 18 from males. A suitably scaled colormap has been used to highlight grey-scale intensity differences.

The grey-scale differences, which are invariant across all the sample faces are strikingly apparent. The eye-eyebrow area seem to always contain dark intensity (low) gray-levels while nose forehead and cheeks contain bright intensity (high) grey-levels. After a great deal of experimentation, the researcher found that the following areas of the human face were suitable for a face detection system based on image invariants and a deformable template.

The above facial area performs well as a basis for a face template, probably because of the clear divisions of the bright intensity invariant area by the dark intensity invariant regions. Once this pixel area is located by the face detection system, any particular area required can be segmented based on the proportions of the average human face After studying the above images it was subjectively decided by the author to use the following as a basis for dark intensity sensitive and bright intensity sensitive templates. Once these are located in a subject's face, a pixel area 33.3% (of the width of the square window).

Note the slight differences which were made to the bright intensity invariant sensitive template which were needed because of the pre-processing done by Department of ECE the system to overcome irregular lighting. Now that a suitable dark and bright intensity invariant templates have been decided on, it is necessary to find a way of using these to make 2 A-units for a perceptron, i.e. a computational model is needed to assign neurons to the distributions displayed

4.5 Face Recognition Process

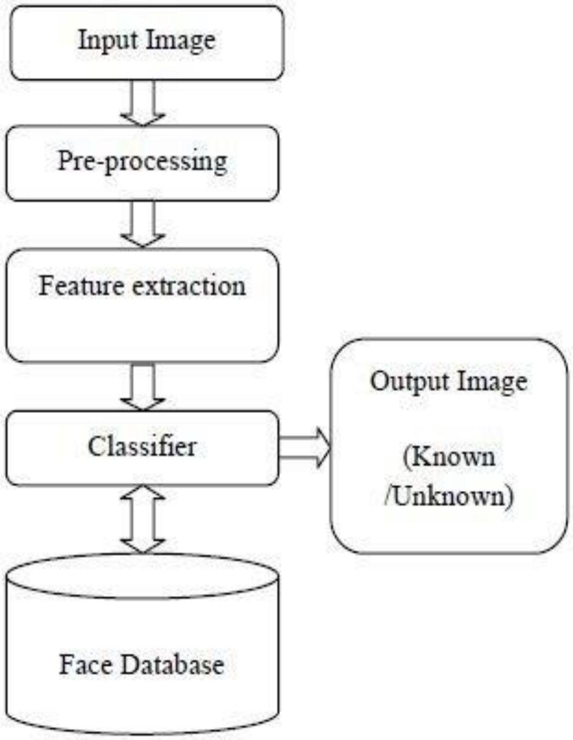


Figure 3. Block Diagram

4.5.1 Face Detection and Extraction: Face detection is important as the image taken through the camera given to the system, face detection algorithm applies to identify the human faces in that image, the number of image processing algorithms are introduce to detect faces in an images and also the location of that detected faces. We have used HOG method to detect human faces in given image.

4.5.2 Face Positioning: There are 68 specific points in a human face. In other words we can say 68 face landmarks. The main function of this step is to detect landmarks of faces and to position the image. A python script is used to automatically detect the face landmarks and to position the face as much as possible without distorting the image.

4.5.3 Face Encoding: Once the faces are detected in the given image, the next step is to extract the unique identifying facial feature for each image. Basically whenever we get localization of face, the 128 key facial point are extracted for each image given input which are highly accurate and these 128-d facial points are stored in data file for face recognition.

4.5.4 Face matching: This is last step of face recognition process. We have used the one of the best learning technique that is deep metric learning which is highly accurate and capable of outputting real value feature vector. Our system ratifies the faces, constructing the 128- d embedding (ratification) for each. Internally compare_faces function is used to compute the Euclidean distance between face in image and all faces in the dataset. If the current image is matched with the 60% threshold with the existing dataset, it will move to attendance marking.

4.6 Attendance Marking

Once the face is identify with the image stored in file, python generate roll numbers of present students and return that, when data is returned, the system generates attendance table which includes the name, roll number, date, day and time with corresponding subject id. And then passes the data to python to store the table into an excel sheet automatically. Each sheet is saved according to the subjects which already entered by the administrator, for example when system generates excel sheet by sending the compiled sheet in an array to python, the python first checks whether there exit any excel sheet of that date, if yes then it create separate worksheet by subject id, so that attendance is differentiated for different subjects.

5. Flow Chart

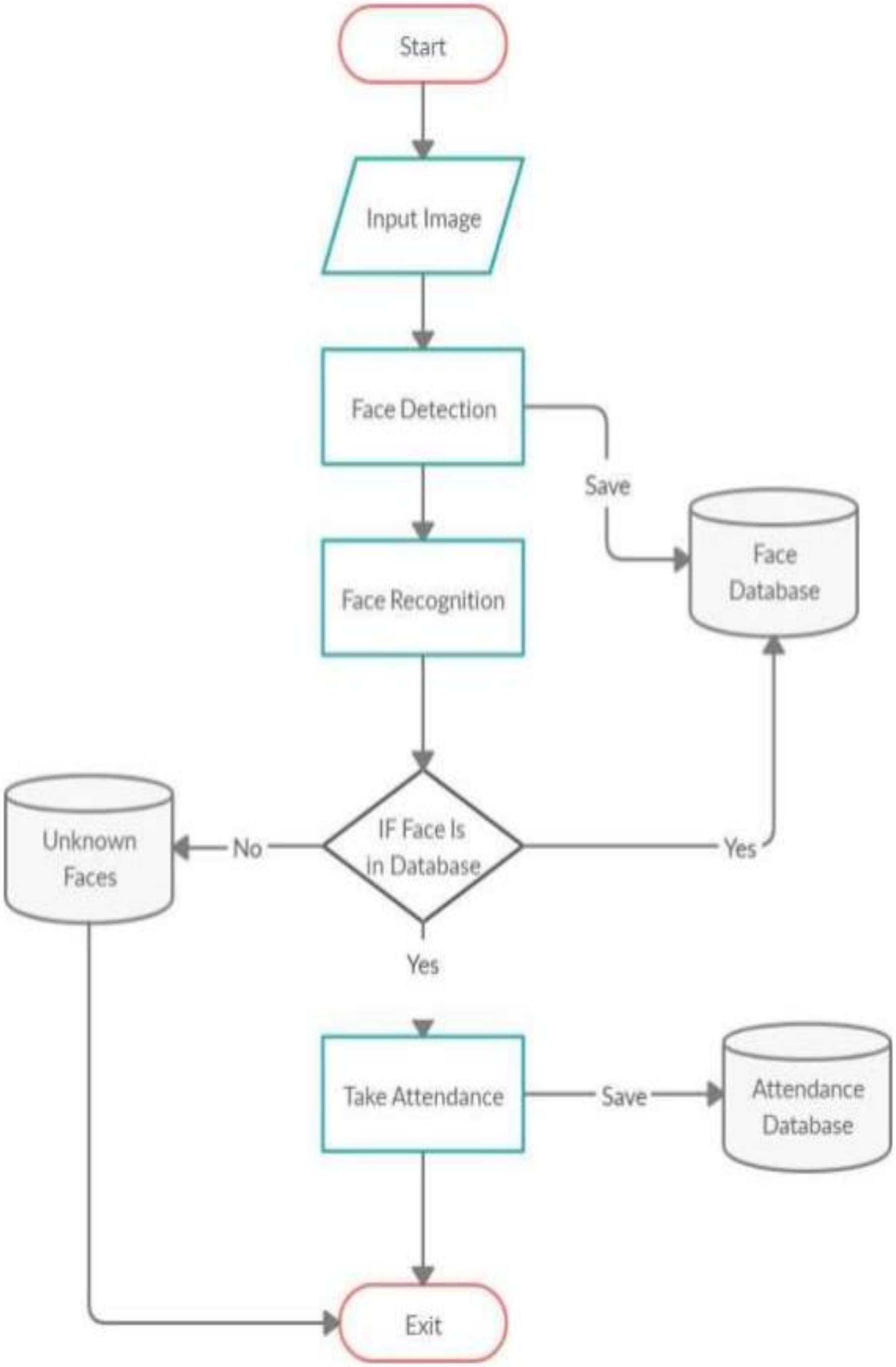


Figure 4. ER-Diagram

6.Implementation

6.1 Code

```
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox as mess
import tkinter.simpledialog as tsd
import cv2,os
import csv
import numpy as np
from PIL import Image
import pandas as pd
import datetime
import time

def assure_path_exists(path):
dir = os.path.dirname(path)
    if not os.path.exists(dir):
os.makedirs(dir)

def tick():
time_string = time.strftime('%H:%M:%S')
clock.config(text=time_string)
clock.after(200,tick)
def contact():
mess._show(title='Contact us', message="Please contact us on :
'xxxxxxxxxxxxx@gmail.com' ")
def check_haarcascadefile():
    exists = os.path.isfile("haarcascade_frontalface_default.xml")
    if exists:
        pass
    else:
```

```
mess._show(title='Some file missing', message='Please contact us for help')
window.destroy()

def save_pass():
assure_path_exists("TrainingImageLabel/")
    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")
    if exists1:
tf = open("TrainingImageLabel\psd.txt", "r")
        key = tf.read()
    else:
master.destroy()
new_pas = tsd.askstring('Old Password not found', 'Please enter a new password below', show='*')
        if new_pas == None:
mess._show(title='No Password Entered', message='Password not set!! Please try again')
        else:
tf = open("TrainingImageLabel\psd.txt", "w")
tf.write(new_pas)
mess._show(title='Password Registered', message='New password was registered successfully!!')
            return
    op = (old.get())
newp= (new.get())
nnewp = (nnew.get())
    if (op == key):
if(newp == nnewp):
txf = open("TrainingImageLabel\psd.txt", "w")
txf.write(newp)
        else:
mess._show(title='Error', message='Confirm new password again!!!')
            return
    else:
mess._show(title='Wrong Password', message='Please enter correct old password.')
        return
mess._show(title='Password Changed', message='Password changed successfully!!')
master.destroy()
def change_pass():
    global master
    master = tk.Tk()
master.geometry("400x160")
master.resizable(False,False)
master.title("Change Password")
master.configure(background="white")
    lbl4 = tk.Label(master,text='    Enter Old Password',bg='white',font=('times', 12, ' bold '))
    lbl4.place(x=10,y=10)
    global old
    old=tk.Entry(master,width=25 ,fg="black",relief='solid',font=('times', 12, ' bold '),show='*')
old.place(x=180,y=10)
    lbl5 = tk.Label(master, text='   Enter New Password', bg='white', font=('times', 12, ' bold '))
    lbl5.place(x=10, y=45)
    global new
    new = tk.Entry(master, width=25, fg="black",relief='solid', font=('times', 12, ' bold '),show='*')
new.place(x=180, y=45)
    lbl6 = tk.Label(master, text='Confirm New Password', bg='white', font=('times', 12, ' bold '))
    lbl6.place(x=10, y=80)
    global nnew
```

```
nnew = tk.Entry(master, width=25, fg="black", relief='solid',font=('times', 12, ' bold '),show='*')
nnew.place(x=180, y=80)
    cancel=tk.Button(master,text="Cancel", command=master.destroy ,fg="black" ,bg="red" ,height=1,width=25 , activebackground = "white" ,font=('times', 10, ' bold '))
cancel.place(x=200, y=120)
    save1 = tk.Button(master, text="Save", command=save_pass, fg="black", bg="#3ece48", height = 1,width=25, activebackground="white", font=('times', 10, ' bold '))
    save1.place(x=10, y=120)
master.mainloop()
def psw():
assure_path_exists("TrainingImageLabel/")
    exists1 = os.path.isfile("TrainingImageLabel\psd.txt")
    if exists1:
tf = open("TrainingImageLabel\psd.txt", "r")
        key = tf.read()
    else:
new_pas = tsd.askstring('Old Password not found', 'Please enter a new password below', show='*')
        if new_pas == None:
mess._show(title='No Password Entered', message='Password not set!! Please try again')
        else:
tf = open("TrainingImageLabel\psd.txt", "w")
tf.write(new_pas)
mess._show(title='Password Registered', message='New password was registered successfully!!')
            return
    password = tsd.askstring('Password', 'Enter Password', show='*')
    if (password == key):
TrainImages()
elif (password == None):
        pass
    else:
mess._show(title='Wrong Password', message='You have entered wrong password')
def clear():
txt.delete(0, 'end')
    res = "1)Take Images  >>>  2)Save Profile"
    message1.configure(text=res)

def clear2():
    txt2.delete(0, 'end')
    res = "1)Take Images  >>>  2)Save Profile"
    message1.configure(text=res)
def TakeImages():
check_haarcascadefile()
    columns = ['SERIAL NO.', '', 'ID', '', 'NAME']
assure_path_exists("StudentDetails/")
assure_path_exists("TrainingImage/")
    serial = 0
    exists = os.path.isfile("StudentDetails\StudentDetails.csv")
    if exists:
        with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:
            reader1 = csv.reader(csvFile1)
            for l in reader1:
                serial = serial + 1
        serial = (serial // 2)
        csvFile1.close()
    else:
        with open("StudentDetails\StudentDetails.csv", 'a+') as csvFile1:
            writer = csv.writer(csvFile1)
writer.writerow(columns)
            serial = 1
        csvFile1.close()
    Id = (txt.get())
    name = (txt2.get())
    if ((name.isalpha()) or (' ' in name)):
        cam = cv2.VideoCapture(0)
harcascadePath = "haarcascade_frontalface_default.xml"
        detector = cv2.CascadeClassifier(harcascadePath)
sampleNum = 0
        while (True):
            ret, img = cam.read()
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            faces = detector.detectMultiScale(gray, 1.3, 5)
            for (x, y, w, h) in faces:
                cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
                # incrementing sample number
sampleNum = sampleNum + 1
                # saving the captured face in the dataset folder TrainingImage
                cv2.imwrite("TrainingImage\ " + name + "." + str(serial) + "." + Id + '.' +
str(sampleNum) + ".jpg",
gray[y:y + h, x:x + w])
                # display the frame
                cv2.imshow('Taking Images', img)
            # wait for 100 miliseconds
            if cv2.waitKey(100) & 0xFF == ord('q'):
                break
            # break if the sample number is morethan 100
elifsampleNum> 100:
                break
cam.release()
        cv2.destroyAllWindows()
        res = "Images Taken for ID : " + Id
        row = [serial, '', Id, '', name]
        with open('StudentDetails\StudentDetails.csv', 'a+') as csvFile:
            writer = csv.writer(csvFile)
writer.writerow(row)
csvFile.close()
        message1.configure(text=res)
    else:
        if (name.isalpha() == False):
            res = "Enter Correct name"
message.configure(text=res)
def TrainImages():
check_haarcascadefile()
assure_path_exists("TrainingImageLabel/")
    recognizer = cv2.face_LBPHFaceRecognizer.create()
harcascadePath = "haarcascade_frontalface_default.xml"
    detector = cv2.CascadeClassifier(harcascadePath)
    faces, ID = getImagesAndLabels("TrainingImage")
    try:
recognizer.train(faces, np.array(ID))
    except:
mess._show(title='No Registrations', message='Please Register someone first!!!')
        return
recognizer.save("TrainingImageLabel\Trainner.yml")
    res = "Profile Saved Successfully"
    message1.configure(text=res)
message.configure(text='Total Registrations till now  : ' + str(ID[0]))
def getImagesAndLabels(path):
    # get the path of all the files in the folder
imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    # createempth face list
    faces = []
    # create empty ID list
    Ids = []
    # now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePaths:
        # loading the image and converting it to gray scale
pilImage = Image.open(imagePath).convert('L')
        # Now we are converting the PIL image into numpy array
imageNp = np.array(pilImage, 'uint8')
        # getting the Id from the image
        ID = int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
faces.append(imageNp)
Ids.append(ID)
    return faces, Ids
def TrackImages():
check_haarcascadefile()
assure_path_exists("Attendance/")
assure_path_exists("StudentDetails/")
for k in tv.get_children():
tv.delete(k)
msg = ''
i = 0
    j = 0
    recognizer        =        cv2.face.LBPHFaceRecognizer_create()              #
cv2.createLBPHFaceRecognizer()
    exists3 = os.path.isfile("TrainingImageLabel\Trainner.yml")
    if exists3:
recognizer.read("TrainingImageLabel\Trainner.yml")
    else:
mess._show(title='Data Missing', message='Please click on Save Profile to reset data!!')
        return
```

```
harcascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(harcascadePath);

    cam = cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_SIMPLEX
col_names = ['Id', '', 'Name', '', 'Date', '', 'Time']
    exists1 = os.path.isfile("StudentDetails\StudentDetails.csv")
    if exists1:
df = pd.read_csv("StudentDetails\StudentDetails.csv")
    else:
mess._show(title='Details Missing', message='Students details are missing, please check!')
cam.release()
        cv2.destroyAllWindows()
window.destroy()
    while True:
        ret, im = cam.read()
gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
        faces = faceCascade.detectMultiScale(gray, 1.2, 5)
        for (x, y, w, h) in faces:
            cv2.rectangle(im, (x, y), (x + w, y + h), (225, 0, 0), 2)
            serial, conf = recognizer.predict(gray[y:y + h, x:x + w])
            if (conf < 50):
ts = time.time()
                date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
                aa = df.loc[df['SERIAL NO.'] == serial]['NAME'].values
                ID = df.loc[df['SERIAL NO.'] == serial]['ID'].values
                ID = str(ID)
                ID = ID[1:-1]
                bb = str(aa)
                bb = bb[2:-2]
                attendance = [str(ID), '', bb, '', str(date), '', str(timeStamp)]

            else:
                Id = 'Unknown'
                bb = str(Id)
            cv2.putText(im, str(bb), (x, y + h), font, 1, (255, 255, 255), 2)
        cv2.imshow('Taking Attendance', im)
        if (cv2.waitKey(1) == ord('q')):
            break
ts = time.time()
    date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
    exists = os.path.isfile("Attendance\Attendance_" + date + ".csv")
    if exists:
        with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:
            writer = csv.writer(csvFile1)
writer.writerow(attendance)
        csvFile1.close()
    else:
        with open("Attendance\Attendance_" + date + ".csv", 'a+') as csvFile1:
            writer = csv.writer(csvFile1)
writer.writerow(col_names)
writer.writerow(attendance)
        csvFile1.close()
    with open("Attendance\Attendance_" + date + ".csv", 'r') as csvFile1:
        reader1 = csv.reader(csvFile1)
        for lines in reader1:
i = i + 1
            if (i> 1):
                if (i % 2 != 0):
iidd = str(lines[0]) + '   '
tv.insert('', 0, text=iidd, values=(str(lines[2]), str(lines[4]), str(lines[6])))
```

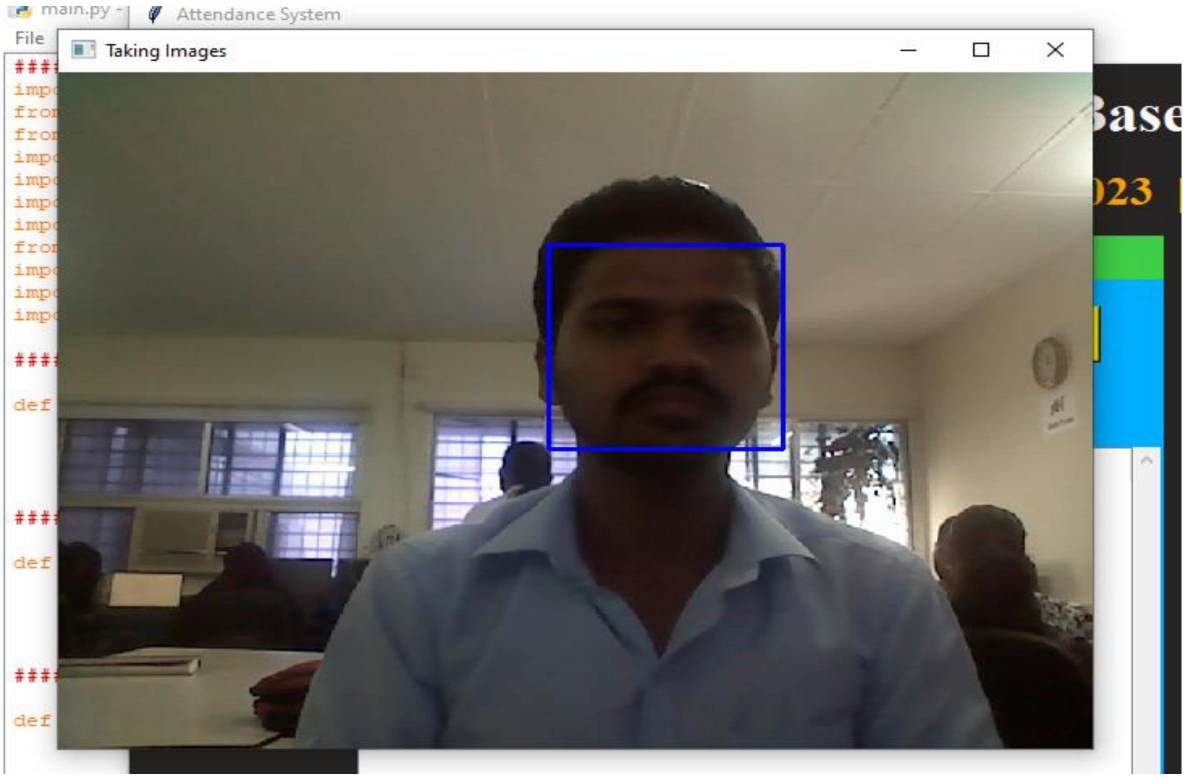
```
    csvFile1.close()
cam.release()
    cv2.destroyAllWindows()
global key
key = ''

ts = time.time()
date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
day,month,year=date.split("-")

mont={'01':'January',
      '02':'February',
      '03':'March',
      '04':'April',
      '05':'May',
      '06':'June',
      '07':'July',
      '08':'August',
      '09':'September',
      '10':'October',
      '11':'November',
      '12':'December'
      }
window = tk.Tk()
window.geometry("1280x720")
window.resizable(True,False)
window.title("Attendance System")
window.configure(background='#262523')

frame1 = tk.Frame(window, bg="#00aeff")
frame1.place(relx=0.11, rely=0.17, relwidth=0.39, relheight=0.80)
```

```
frame2 = tk.Frame(window, bg="#00aeff")
frame2.place(relx=0.51, rely=0.17, relwidth=0.38, relheight=0.80)

message3 = tk.Label(window, text="Face Recognition Based Attendance System"
,fg="white",bg="#262523" ,width=55 ,height=1,font=('times', 29, ' bold '))
message3.place(x=10, y=10)

frame3 = tk.Frame(window, bg="#c4c6ce")
frame3.place(relx=0.52, rely=0.09, relwidth=0.09, relheight=0.07)

frame4 = tk.Frame(window, bg="#c4c6ce")
frame4.place(relx=0.36, rely=0.09, relwidth=0.16, relheight=0.07)

datef = tk.Label(frame4, text = day+"-"+mont[month]+"-"+year+"  |  ",
fg="orange",bg="#262523" ,width=55 ,height=1,font=('times', 22, ' bold '))
datef.pack(fill='both',expand=1)

clock = tk.Label(frame3,fg="orange",bg="#262523" ,width=55 ,height=1,font=('times', 22,
' bold '))
clock.pack(fill='both',expand=1)
tick()

head2 = tk.Label(frame2, text="                       For New Registrations                       ",
fg="black",bg="#3ece48" ,font=('times', 17, ' bold ') )
head2.grid(row=0,column=0)

head1 = tk.Label(frame1, text="                       For Already Registered                       ",
fg="black",bg="#3ece48" ,font=('times', 17, ' bold ') )
head1.place(x=0,y=0)
```

```
lbl = tk.Label(frame2, text="Enter ID",width=20  ,height=1  ,fg="black"  ,bg="#00aeff"
,font=('times', 17, ' bold ') )
lbl.place(x=80, y=55)

txt = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold '))
txt.place(x=30, y=88)

lbl2 = tk.Label(frame2, text="Enter Name",width=20  ,fg="black"  ,bg="#00aeff"
,font=('times', 17, ' bold '))
lbl2.place(x=80, y=140)

txt2 = tk.Entry(frame2,width=32 ,fg="black",font=('times', 15, ' bold ')  )
txt2.place(x=30, y=173)

message1 = tk.Label(frame2, text="1)Take Images          2)Save Profile" ,bg="#00aeff"
,fg="black"  ,width=39 ,height=1, activebackground = "yellow" ,font=('times', 15, ' bold '))
message1.place(x=7, y=230)

message = tk.Label(frame2, text="" ,bg="#00aeff" ,fg="black"  ,width=39,height=1,
activebackground = "yellow" ,font=('times', 16, ' bold '))
message.place(x=7, y=450)

lbl3 = tk.Label(frame1, text="Attendance",width=20  ,fg="black"  ,bg="#00aeff"
,height=1 ,font=('times', 17, ' bold '))
lbl3.place(x=100, y=115)

res=0
exists = os.path.isfile("StudentDetails\StudentDetails.csv")
if exists:
    with open("StudentDetails\StudentDetails.csv", 'r') as csvFile1:
        reader1 = csv.reader(csvFile1)
        for l in reader1:
            res = res + 1
    res = (res // 2) - 1
    csvFile1.close()
else:
    res = 0
message.configure(text='Total Registrations till now  : '+str(res))
menubar = tk.Menu(window,relief='ridge')
filemenu = tk.Menu(menubar,tearoff=0)
filemenu.add_command(label='Change Password', command = change_pass)
filemenu.add_command(label='Contact Us', command = contact)
filemenu.add_command(label='Exit',command = window.destroy)
menubar.add_cascade(label='Help',font=('times', 29, ' bold '),menu=filemenu)
tv= ttk.Treeview(frame1,height =13,columns = ('name','date','time'))
tv.heading('#0',text ='ID')
tv.heading('name',text ='NAME')
tv.heading('date',text ='DATE')
tv.heading('time',text ='TIME')
tv.grid(row=2,column=0,padx=(0,0),pady=(150,0),columnspan=4)
tv.column('#0',width=82)
tv.column('name',width=130)
tv.column('date',width=133)
tv.column('time',width=133
scroll=ttk.Scrollbar(frame1,orient='vertical',command=tv.yview)
scroll.grid(row=2,column=4,padx=(0,100),pady=(150,0),sticky='ns')
tv.configure(yscrollcommand=scroll.set)
clearButton = tk.Button(frame2, text="Clear", command=clear ,fg="black"
,bg="#ea2a2a"  ,width=11 ,activebackground = "white" ,font=('times', 11, ' bold '))
clearButton.place(x=335, y=86)
clearButton2 = tk.Button(frame2, text="Clear", command=clear2 ,fg="black"
,bg="#ea2a2a"  ,width=11 , activebackground = "white" ,font=('times', 11, ' bold '))
```

```
clearButton2.place(x=335, y=172)
takeImg = tk.Button(frame2, text="Take Images", command=TakeImages  ,fg="white"  ,bg="Magenta"  ,width=34  ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))
takeImg.place(x=30, y=300)
trainImg = tk.Button(frame2, text="Save Profile", command=psw  ,fg="white"  ,bg="Magenta"  ,width=34  ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))
trainImg.place(x=30, y=380)
trackImg = tk.Button(frame1, text="Take Attendance", command=TrackImages  ,fg="black"  ,bg="yellow"  ,width=35  ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))
trackImg.place(x=30,y=50)
quitWindow = tk.Button(frame1, text="Quit", command=window.destroy  ,fg="black"  ,bg="red"  ,width=35 ,height=1, activebackground = "white" ,font=('times', 15, ' bold '))
quitWindow.place(x=30, y=450)
window.configure(menu=menubar)
window.mainloop()
```

7.Results



```
####
impo
from
from
impo
impo
impo
impo
from
impo
impo
impo

####

def



####

def



####

def
```

Figure 5. Dataset Capture

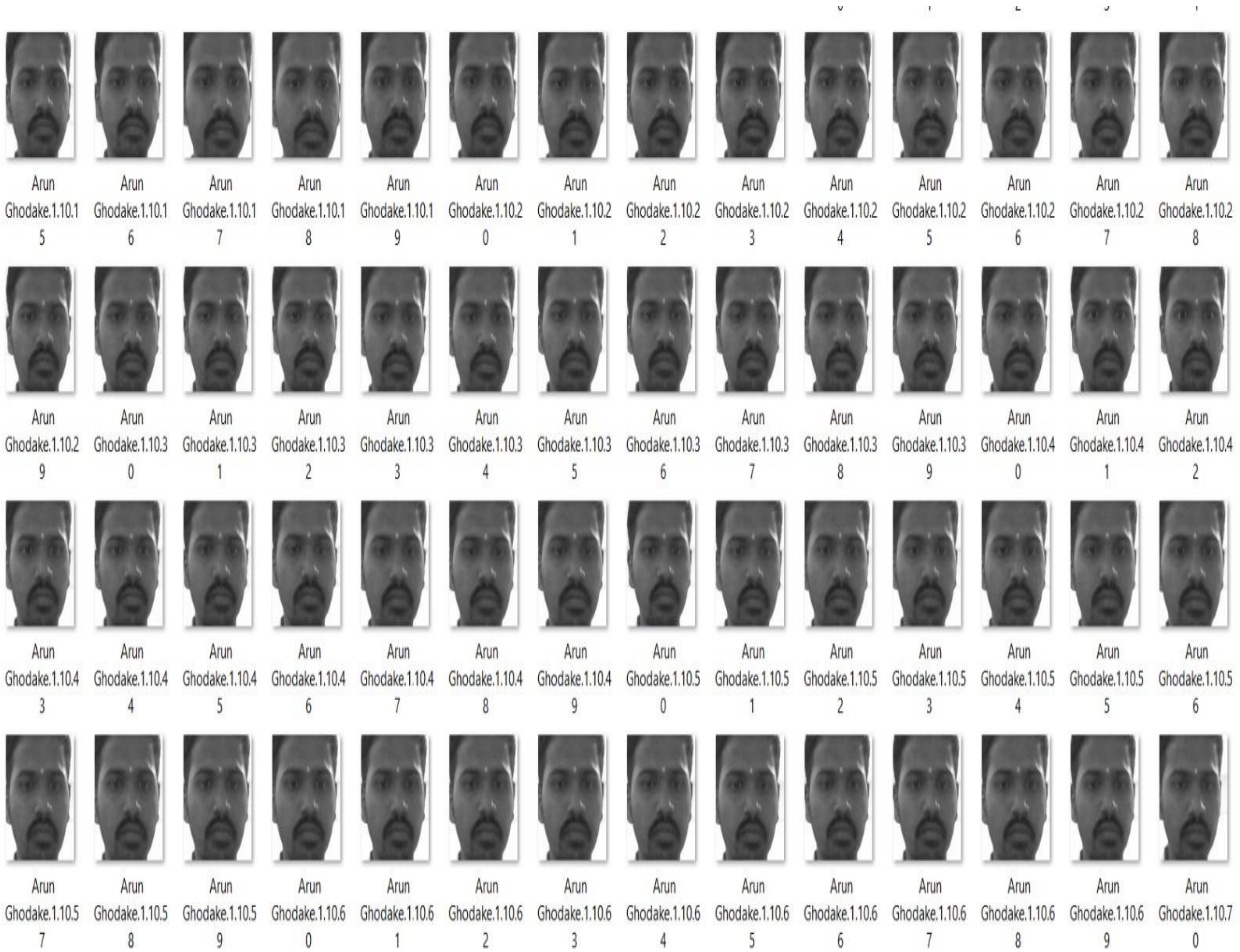


Figure 6. Database Sample

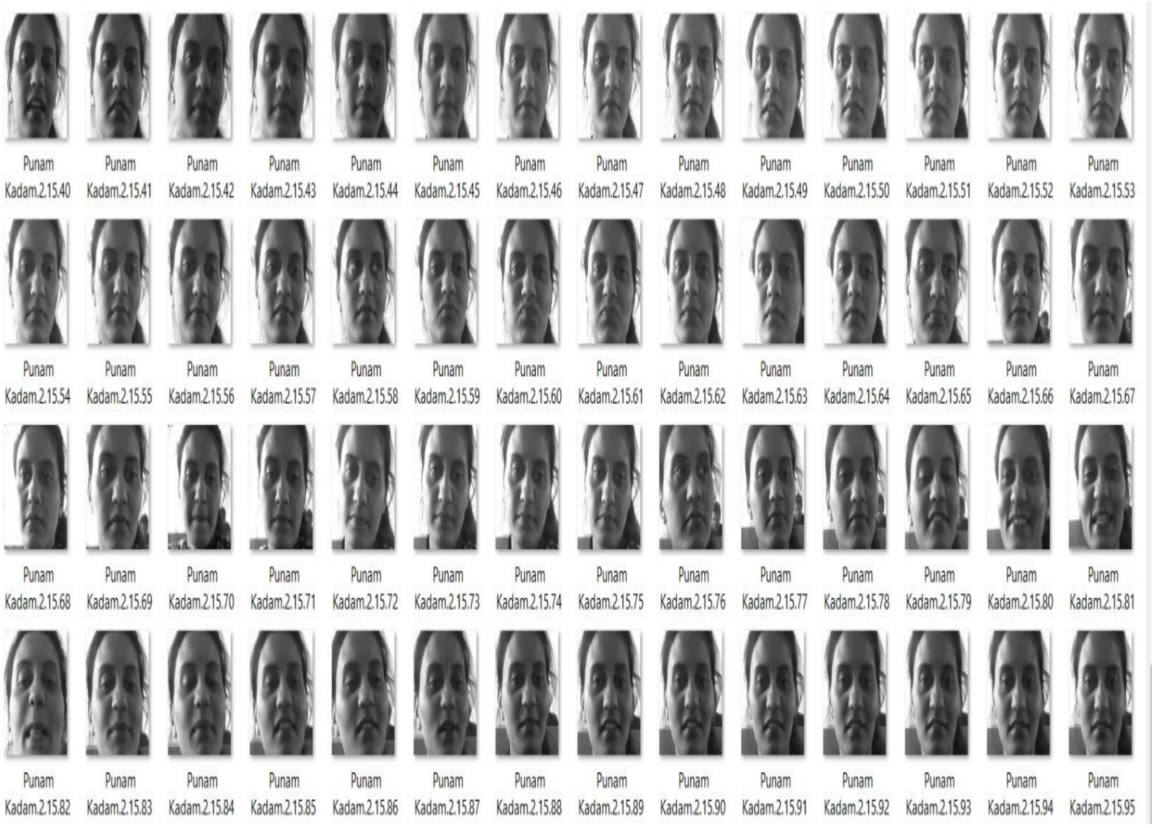
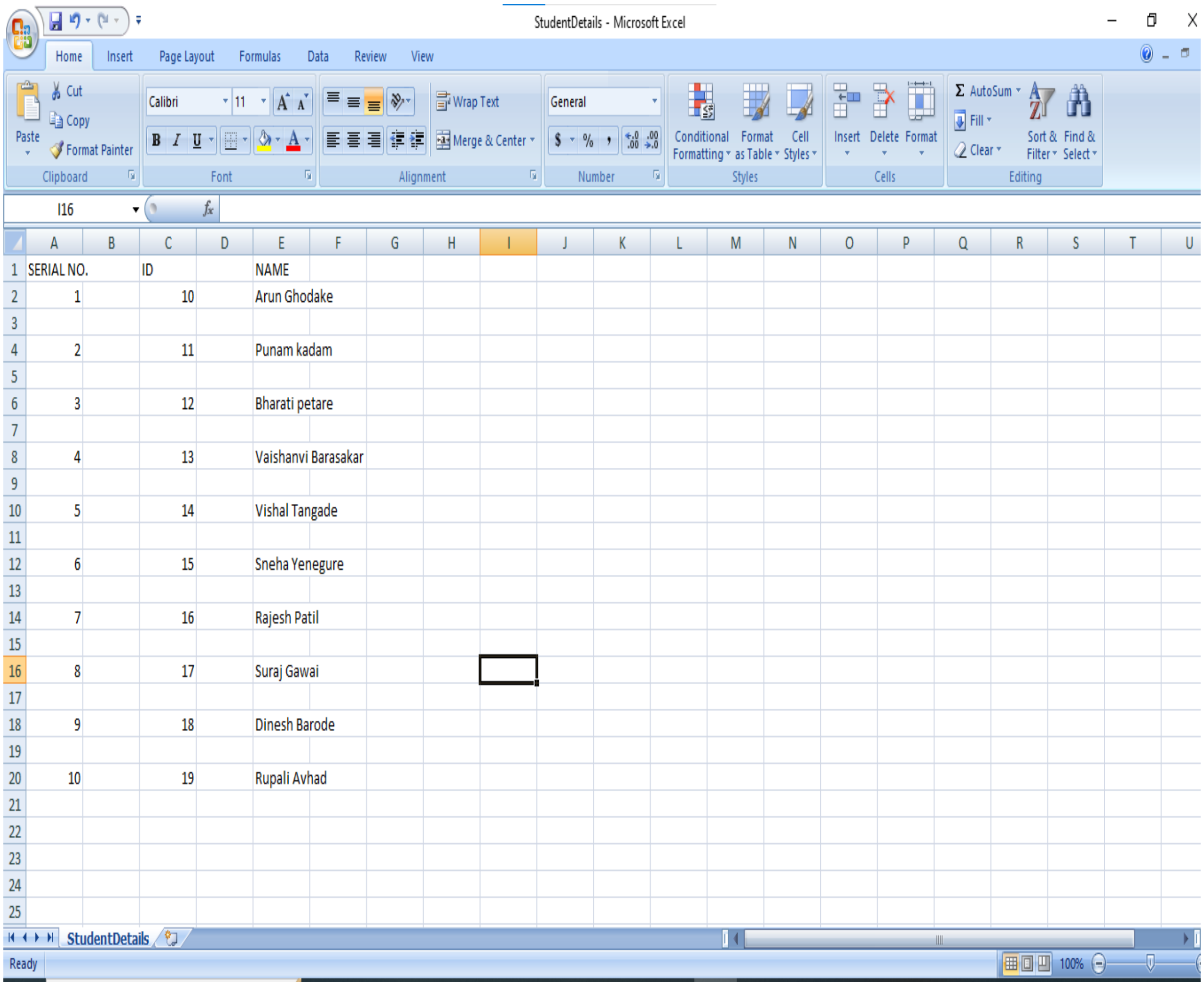
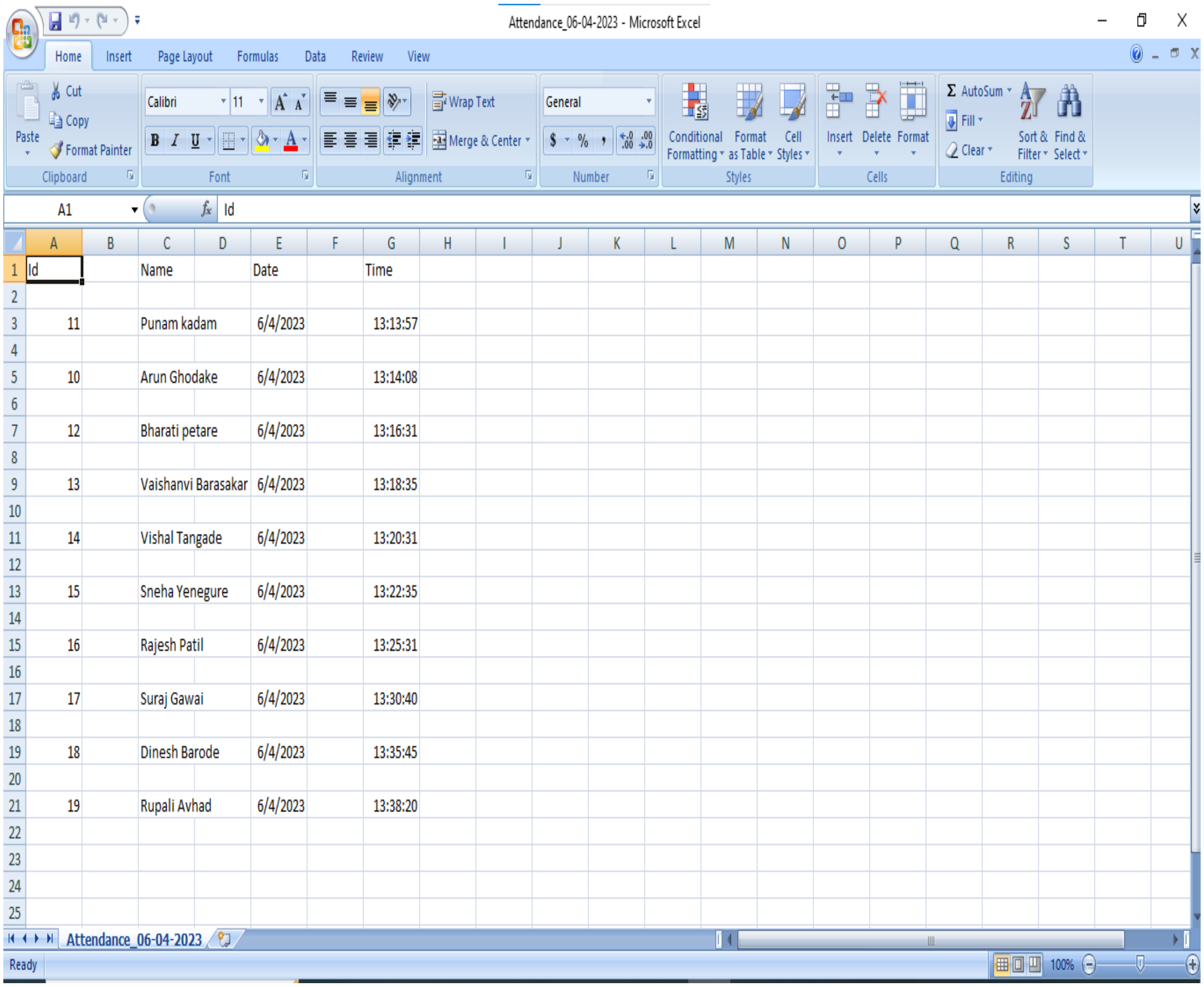


Figure 7. Database Sample



	A	B	C	D	E
1	SERIAL NO.		ID		NAME
2	1		10		Arun Ghodake
3					
4	2		11		Punam kadam
5					
6	3		12		Bharati petare
7					
8	4		13		Vaishanvi Barasakar
9					
10	5		14		Vishal Tangade
11					
12	6		15		Sneha Yenegure
13					
14	7		16		Rajesh Patil
15					
16	8		17		Suraj Gawai
17					
18	9		18		Dinesh Barode
19					
20	10		19		Rupali Avhad

Figure 8 Excel File 1.1 (Student Details)



Id	Name	Date	Time
11	Punam kadam	6/4/2023	13:13:57
10	Arun Ghodake	6/4/2023	13:14:08
12	Bharati petare	6/4/2023	13:16:31
13	Vaishanvi Barasakar	6/4/2023	13:18:35
14	Vishal Tangade	6/4/2023	13:20:31
15	Sneha Yenegure	6/4/2023	13:22:35
16	Rajesh Patil	6/4/2023	13:25:31
17	Suraj Gawai	6/4/2023	13:30:40
18	Dinesh Barode	6/4/2023	13:35:45
19	Rupali Avhad	6/4/2023	13:38:20

Figure 9 Excel File1.2 (Attendance Sheet)

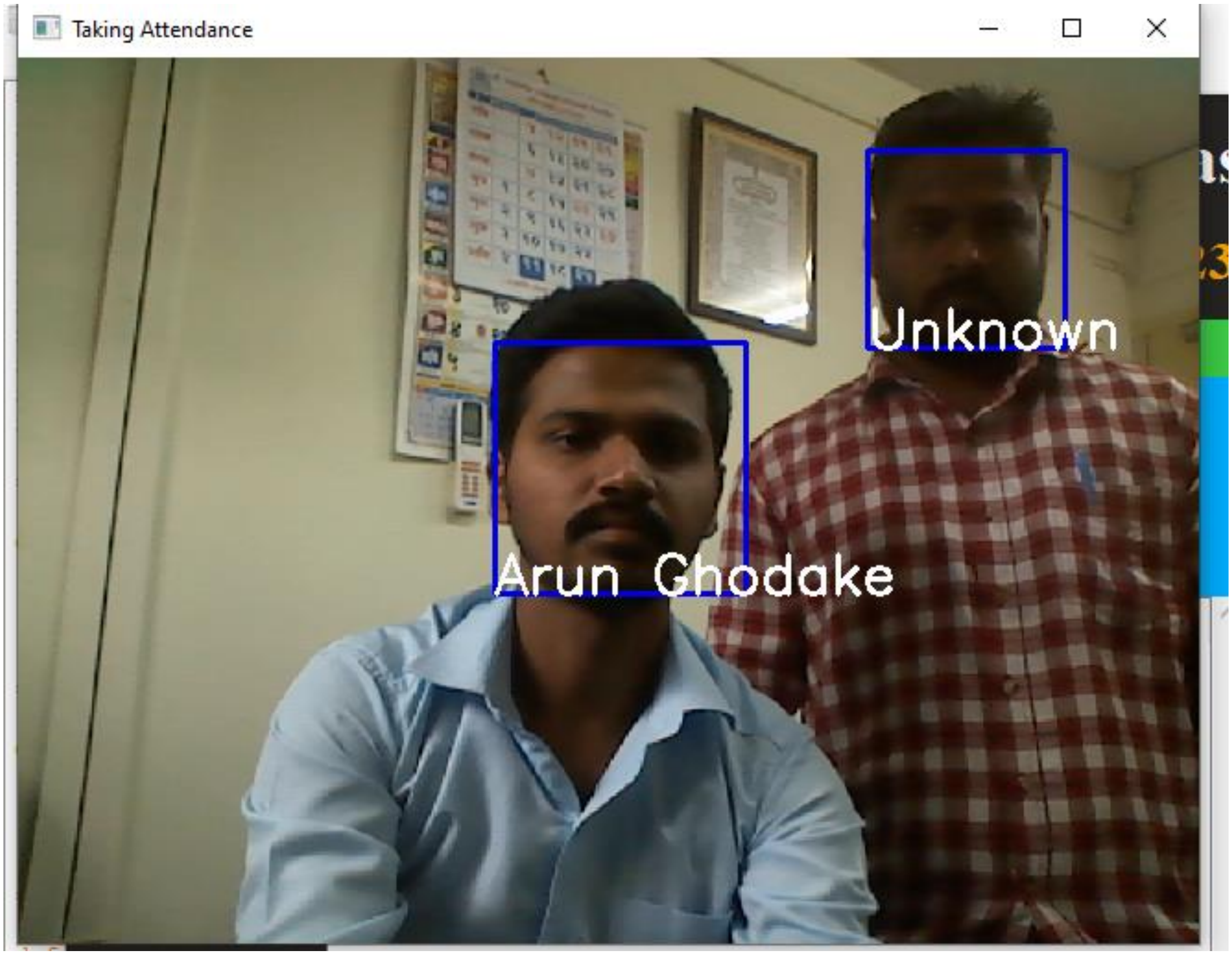


Figure 10.Attendance Marking

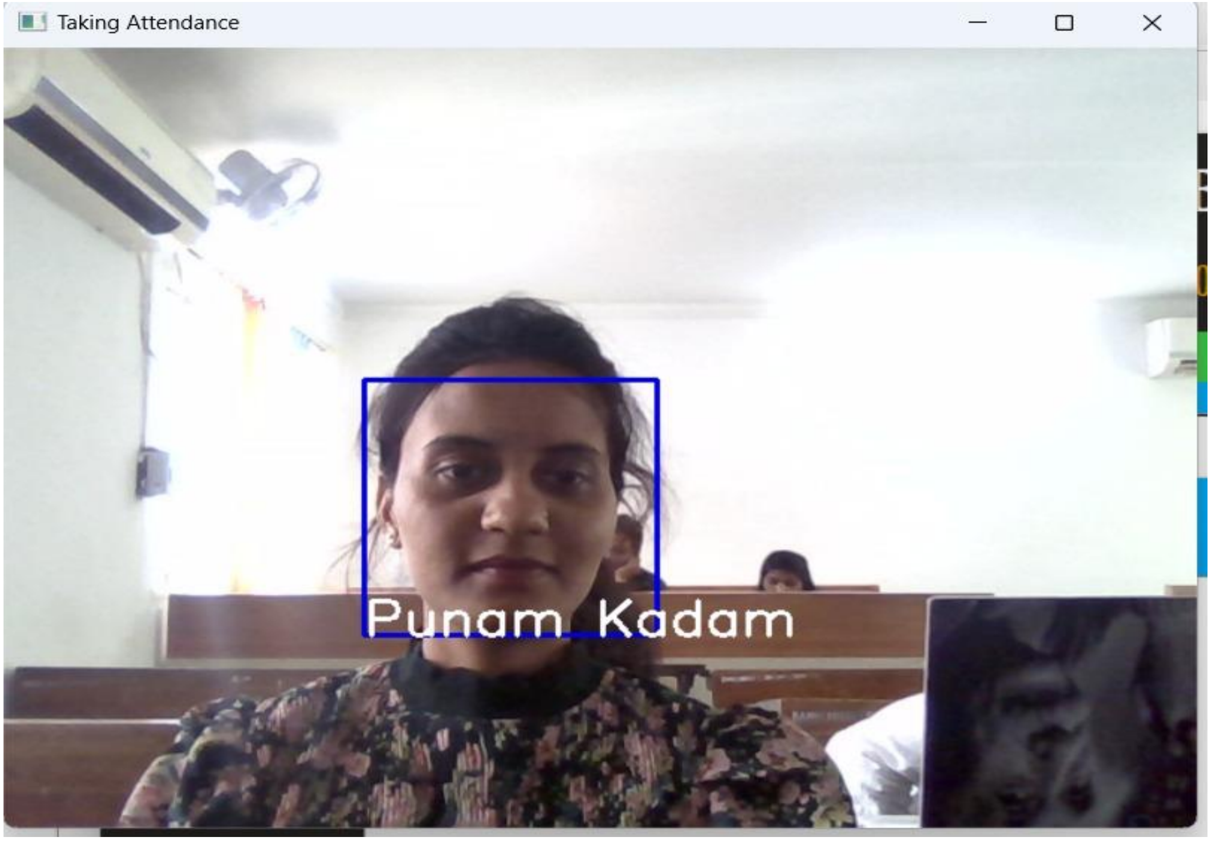


Figure 11.Attendance Marking

8. Conclusion

- We have presented a face detector with a reasonably good accuracy and running time. However, many aspects of the design are tuned for the constrained scene conditions of the training images provided, hurting its robustness.

- This is not unfair given the scope and requirements of the project. Our algorithm is sensitive to the color information in the image and will not work for a gray scale image.

- We feel that detecting connected faces was the hardest part of the project. A great deal of time was spent coming up with a template matching scheme that adapts well to connected faces, including those that are partly visible.

9. Future Scope

- Using this system we will able to accomplish the task of marking the attendance in the classroom automatically and output is obtained in an excel sheet as desired in real-time

- However, in order to develop a dedicated system which can be implemented in an educational institution, a very efficient algorithm which is insensitive to the lighting conditions of the classroom has to be developed.

- Another important aspect where we can work towards is creating an online database of the attendance and automatic updating of the attendance.

References

1. Pandey, Anurag and Choudhary, Divyansh and Agarwal, Ritik and Shrivastava, Tushar and, Kriti,“Face detection using Haar cascade classifier” (July 14, 2022). Proceedings of the Advancement in Electronics & Communication Engineering 2022, https://ssrn.com
2. Dr. V Suresh, Srinivasa ChakravarthiDumpa, Chiranjeevi Deepak Vankayala, HaneeshaAduri, Jayasree Rapa,” Facial Recognition Attendance System Using Python and OpenCv”(25Feb , 2020),www.questjournals.org
3. Yohei KAWAGUCHI Tetsuo SHOJI WeijaneLIN Koh KAKUSHO Michihiko MINOH,“FaceRecognition-based Lecture Attendance System” https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=4b6811cd2a7a6924fed4967c2b755c0942ca5351
4. K.Senthamil Selvi1 , P.Chitrakala2 , A.Antony Jenitha3,“ face recognition based attendance marking system”(February 2014,)www.ijcsmc.com
5. Naveed Khan Balcoh, M. Haroon Yousaf, Waqar Ahmad and M. Iram Baig
   a. “ Algorithm for Efficient Attendance Management: Face Recognition based approach” (July 2012,)www.IJCSI.org
6. PAUL VIOLA Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA MICHAEL J. JONES Mitsubishi Electric Research Laboratory, 201 Broadway, Cambridge, MA 02139, USA,“Robust Real-Time Face Detection”
7. Sovitkar Sarika. A, Kawathekar Seema. S, “A Study of Hybrid Approach for Face Recognition Using Student Database,” InMachine Learning for Predictive Analysis: Proceedings of ICTIS 2020 2021 (pp. 375-383). Springer Singapore.
8. Hassen O. A, Kh-Madhloom. J, Ali. A. A, Abidin. Z. Z, “Face Smile Detection and Predictive Recognition Using Binary Locality Preserving Projections”, Jour of Adv Research in Dynamical & Control Systems, Vol. 10, 10-Special Issue, 2018
9. Septi SQ, Yulita IN, Napitupulu H, “Face Recognition Using Fisherface and Support Vector Machine Method”, In2021 International Conference on Artificial Intelligence and Big Data Analytics 2021 Oct 27 (pp. 50-55). IEEE.
10. Huang YH, Fuh CS, "Face detection and smile detection,” InProceedings of IPPR Conference on Computer Vision, Graphics, and Image Processing (CVGIP) 2009.

11. Choi CH, Kim J, Hyun J, Kim Y, Moon B, "Face detection using haar cascade classifiers based on vertical component calibration", Human-centric Computing and Information Sciences. 2022 Mar 15;12(11).
12. Rahmad C, Asmara RA, Putra DR, Dharma I, Darmono H, Muhiqqin I, "Comparison of Viola-Jones Haar Cascade classifier and histogram of oriented gradients (HOG) for face detection," InIOP conference series: materials science and engineering 2020 (Vol. 732, No. 1, p. 012038). IOP Publishing
13. Batwani K, Sabeer M, "A REVIEW OF FACE DETECTION AND SMILE DETECTION", International Journal of Innovative Science and Research Technology Volume 1 , Issue 2 , May - 2016
14. Varadharajan E, Dharani R, Jeevitha S, Kavinmathi B, Hemalatha S, "Automatic attendance management system using face detection," In2016 Online international conference on green engineering and technologies (IC-GET) 2016 Nov 19 (pp. 1-3). IEEE.
15. Kowsalya P, Pavithra J, Sowmiya G, Shankar CK, "Attendance monitoring system using face detection & face recognition," International Research Journal of Engineering and Technology (IRJET). 2019 Mar; 6(3):6629-32.

Also Visited:

- https://developers.google.com/ml-kit/vision/face-detection/face-detection-concepts
- https://github.com/parulnith/Face-Detection-in-Python-using-OpenCV/tree/master/Face%20Detection
- https://www.analyticsvidhya.com/blog/2021/06/learn-how-to-implement-face-recognition-using-opencv-with-python/
- https://github.com/shubhamkumar27/Face_recognition_based_attendance_system