1. INTRODUCTION

The advancement of technology has brought about new and innovative ways to manage various processes in different industries. One such process is attendance management, which is a crucial aspect of any educational institution or organization. The traditional methods of attendance management are often time-consuming and prone to errors. To overcome these challenges, the development of an automatic facial attendance system using OpenCV and Python has become necessary. This system leverages the power of facial recognition technology to accurately identify and verify individuals in real-time, thereby automating the attendance process. The use of OpenCV, an open-source computer vision library, and Python, a widely-used programming language, ensures that the system is efficient and scalable. The system can be easily integrated into existing attendance management systems and provides attendance records in a digital format, eliminating the need for manual processes. The goal of this project is to upgrade the current attendance system, making it more efficient, effective and accurate. By implementing this system, educational institutions and organizations can streamline their attendance management processes, saving time and reducing the risk of errors.

1.1 INTRODUCTION TO TOOLS

1.1.1 PYTHON

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.Python can be used on a server to create web applications.Python can be used alongside software to create workflows.Python can connect to database systems. It can also read and modify files.Python can be used to handle big data and perform complex mathematics.Python can be used for rapid prototyping, or for production-ready software development.Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).Python has a simple syntax similar to the English language.Python has syntax that allows developers to write programs with fewer lines than some other programming languages.Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.Python can be treated in a procedural way, an object-oriented way or a functional way.Python can be used for rapid prototyping, or for production-ready software development.Python works on different platforms

(Windows, Mac, Linux, Raspberry Pi, etc).Python has a simple syntax similar to the English language.Python has syntax that allows developers to write programs with fewer lines than some other programming languages.Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.Python can be treated in a procedural way, an object- oriented way or a functional way.

1.1.1.1 PYTHON 2 V/S PYTHON 3

Python 3 is more in-demand and includes a typing system. Python 2 is outdated and uses an older syntax for the print function. While Python 2 is still in use for configuration management in DevOps, Python 3 is the current standard. Python (the code, not the snake) is a popular coding language to learn for beginners.

1.1.1.2 PYTHON 3

Python 3 is an improved version. It supports Unicode characters. If we dive some integer, it gives exact output. Earlier in python 2, we get a rounded value for the same. Most of the IT companies switching towards python3. It gets faster with every newer version. Also, Python has big library support for python3. Companies like Facebook, Instagram are using Python 3 as their language. The most important thing to make a note is Python 2.7 will not be supported after 2020.

1.1.2 COMPUTER VISION

Computer vision is a field of study which encompasses on how computer see and understand digital images and videos.Computer vision involves seeing or sensing a visual stimulus, make sense of what it has seen and also extract complex information that could be used for other machine learning activities.Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs — and take actions or make recommendations based on that information.Computer vision is an interdisciplinary scientific field that deals with how computers can gain high-level understanding from digital images or videos.

1.1.2.1 APPLICATIONS OF COMPUTER VISION

- **Autonomous Vehicles** — This is one of the most important applications of Computer vision where the self-driving cars need to gather information about their surroundings to decide how to behave.

- **Facial Recognition —** This is also a very important application of computer vision where electronics use facial recognition technology to basically validate theidentity of the user.

- **Image Search and Object Recognition** — Now we could search objects in an image using image search. A very good example is google lens where we could search a particular object within the image by clicking the photo of the image and the computer vision algorithm will search through the catalogue of images and extract information out of the image.

- **Robotics** — Most robotic machines, often in manufacturing, need to see their surroundings to perform the task at hand. In manufacturing machines may be used to inspect assembly tolerances by "looking at" them.

1.1.2.2 OPENCV

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV. OpenCV was started at Intel in 1999by **Gary Bradsky**, and the first release came out in 2000. **Vadim Pisarevsky** joined Gary Bradsky to manage Intel's Russian software OpenCV team. In 2005, OpenCV was used on Stanley, the vehicle that won the 2005 DARPA Grand Challenge. Later, its active development continued under the support of Willow Garage with Gary Bradsky and Vadim Pisarevsky leading the project. OpenCV now supports a multitude of algorithms related to Computer Vision and Machine Learning and is

expanding day by day. OpenCV supports a wide variety of programming languages such as C++, Python, Java, etc., and is available on different platforms including Windows, Linux, OS X, Android, and iOS. Interfaces for high-speed GPU operations based on CUDA and OpenCL are also under active development.OpenCV-Python is the Python API for OpenCV, combining the best qualities of the OpenCV C++ API and the Python language.

1.1.2. 3 OpenCV-Python

OpenCV-Python is a library of Python bindings designed to solve computer vision problems.Python is a general purpose programming language started by **Guido van Rossum** that became very popular very quickly, mainly because of its simplicity and code readability. It enables the programmer to express ideas in fewer lines of code without reducing readability.Compared to languages like C/C++, Python is slower. That said, Python can be easily extended with C/C++, which allows us to write computationally intensive code in C/C++ and create Python wrappers that can be used as Python modules. This gives us two advantages: first, the code is as fast as the original C/C++ code (since it is the actual C++ code working in background) and second, it easier to code in Python than C/C++. OpenCV-Python is a Python wrapper for the original OpenCV C++ implementation.OpenCV-Python makes use of **Numpy**,which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays.This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.

1.1.2.4 NUMPY

NumPy is a powerful open-source library for Python that provides a set of tools for working with numerical data. At its core, NumPy provides a set of array objects that are designed to make it easy to work with multi-dimensional data. These arrays are similar to lists or tuples, but provide a number of additional features that make them ideal for working with numerical data. One of the key benefits of NumPy is its speed and efficiency. Because it is written in C, NumPy provides a fast and efficient way to work with large datasets. This is particularly important in scientific computing and data analysis, where large datasets are common.

NumPy is widely used in a variety of fields, including science, engineering, finance, and

more. It is also a key component of other popular data analysis libraries in Python.NumPy is an essential tool for anyone working with numerical data in Python. Its powerful set of array objects and functions make it easy to work with large datasets, and its speed and efficiency make it ideal for scientific computing and data analysis.

1.1.2.5 PANDAS

Pandas is a popular data analysis library for Python that provides a powerful and flexible set of tools for working with structured data. It is built on top of the NumPy library, and provides a number of data structures and functions that are designed to make working with data in Python fast and efficient. At the core of Pandas is the DataFrame, a two-dimensional table-like data structure that provides a flexible way to store and manipulate data. The DataFrame is similar to a spreadsheet or SQL table and allows you to work with data in a variety of ways, including filtering, sorting, grouping, merging, and more. It also provides a powerful set of functions for working with missing or null data, as well as time series data. Pandas also provides a number of other data structures, including the Series, which represents a one-dimensional array-like object, and the Panel, which represents a three-dimensional array-like object. These structures are designed to make it easy to work with data in a variety of ways, and provide a powerful set of functions for data manipulation and analysis. Overall, Pandas is an essential tool for anyone working with data in Python. Its powerful set of data structures and functions make it easy to work with structured data, and its integration with other libraries in python

1.1.2.6 YAGMAIL

Yagmail is a Python library that simplifies the process of sending emails from your Python code. It is built on top of the standard Python library for sending emails (smtplib), but provides a simpler and more intuitive interface. Yagmail allows you to send emails with just a few lines of code, and supports sending attachments, HTML content, and even using Gmail aliases.

Yagmail makes it easy to authenticate with your email provider, and can be used with any email service that provides SMTP support. It also provides a number of convenience functions, such as automatically adding a signature to your emails or sending emails to multiple recipients.

Yagmail is a lightweight library with no external dependencies, making it easy to install and use. It is well-documented and actively maintained, with a friendly community of developers providing support and contributions. Yagmail is a useful tool for anyone who needs to send emails from their Python code, and provides a simple and intuitive interface for doing so.

1.1.3 CLASSIFIER

A classifier is a type of machine learning algorithm used to assign a class label to a data input. An example is an image recognition classifier to label an image (e.g., “car,” “truck,” or “person”). Classifier algorithms are trained using labeled data; in the image recognition example, for instance, the classifier receives training data that labels images. After sufficient training, the classifier then can receive unlabeled images as inputs and will output classification labels for each image.

Classifier algorithms employ sophisticated mathematical and statistical methods to generate predictions about the likelihood of a data input being classified in a given way. In the image recognition example, the classifier statistically predicts whether an image is likely to be a car, a truck, or a person, or some other classification that the classifier has been trained to identify.

1.1.3.1 HAAR CASCADE CLASSIFIER

Haar Cascade classifier is a popular computer vision algorithm used for object detection. It was developed by Viola and Jones in 2001 and is based on the Haar wavelet transform. The algorithm uses a machine learning approach to detect objects in images or video streams by first training a classifier on positive and negative images. Positive images contain the object of interest, while negative images do not. The algorithm extracts features from these images using the Haar wavelet transform, which is a mathematical technique for detecting edges and other patterns in images. The classifier is then trained using these features, and can be used to detect objects in new images or video streams. The algorithm works by sliding a window over the image and evaluating the trained classifier at each position to determine whether the object is present in that location. Haar Cascade classifier is widely used for face detection, but can be trained to detect other objects as well. The algorithm is fast and can be used in real-time applications, making it a popular choice for a variety of computer vision tasks. Haar Cascade classifier has been widely adopted in a range of applications, including facial recognition, pedestrian detection, and traffic sign recognition. The algorithm has proved to be effective and robust, and has paved the way for more advanced object detection algorithms.

1.1.4 METHODOLOGY

The methodology for the automatic facial attendance system involves collecting a dataset of facial images, training the facial recognition algorithm using OpenCV, developing the system using Python, integrating it with existing attendance management systems, testing and evaluating the system, and deploying it for attendance management purposes. The combination of OpenCV and Python ensures a robust and scalable system that accurately detects and matches faces in real-time, improving the efficiency and accuracy of attendance taking.

1.1.4.1 PROPOSED WORK-FLOW

We decided to build a very simple and basic Haar cascade classifier algoritham model using numpy library and OpenCV to detect face of students for marking their attendance

1.1.4.2 IMAGE PROCESSING

Haar Cascade Classifier will detect the input from videocam. The images captured by the system's webcam required pre-processing before going to the next step. In the pre-processing step, the image is transformed into a grayscale image because the RGB color image contains so much redundant information that is not necessary for fac detection. Then, we resized the images into (150x150) size to maintain uniformity of the input images to the architecture. Then, the images are normalized and after normalization, the value of a pixel resides in the range from 0 to 1. Normalization helped the learning algorithm to learn faster and captured necessary features from the images.

1.1.4.3 DATASET COLLECTION

To train our deep learning architecture, we collected images. The architecture of the learning technique highly depends on LBPH. Data from source[10] is collected for training and testing the model. Dataset contains images of faces only. It consists of about 101 images for each individual person .The total number of images present

in the dataset is equal to number of people multiplied by 101. For training purpose, 99% of images are used and the rest 1% of the images may have some error. Fig. 2 shows some of the images used for training purpose

1.1.4.4 ARCHITECTURE DEVELOPMENT

The learning model is based on LBPH (Local Binary Patterns Histograms) algorithm.It is a widely used face recognition method that is based on the concept of local binary patterns. These patterns describe the relative intensity of pixels in a given region and can be used to represent the features of a face. The LBPH algorithm operates in several steps, including preprocessing, local binary pattern calculation, histogram calculation, and comparison.In the preprocessing step, the image is converted to grayscale and resized to a common size. The next step involves calculating the local binary patterns for each pixel in the image by considering a small region around each pixel, computing the intensity differences between the center pixel and its neighbors, and encoding the results as a binary pattern.The histogram calculation step involves constructing a histogram to represent the distribution of patterns in the image, which serves as a representation of the face's features. Finally, the histograms of the test image are compared with those of the reference images in the database using a distance metric such as the Euclidean distance or the Chi-squared distance. The reference image with the smallest distance is considered to be the best match. Overall, the LBPH algorithm is simple to implement and has been found to be effective for face recognition tasks. However, it can be sensitive to variations in illumination, scale, and rotation, so these factors must be taken into account when designing the system.

1.1.5 MACHINE LEARNING ARCHITECTURE

The machine learning architecture defines the various layers involved in the machine learning cycle and involves the major steps being carried out in the transformation of raw data into training data sets capable for enabling the decision making of a system. Machine Learning architecture is defined as the subject that has evolved from the concept of fantasy to the proof of reality. As earlier machine learning approach forpattern recognitions has lead foundation for the upcoming major artificial intelligence program. Based upon the different algorithm that is used on the training data machine

learning architecture is categorized into three types i.e. Supervised Learning, Unsupervised Learning, and Reinforcement Learning and the process involved in this architecture are Data Acquisition , Data Processing, Model Engineering, Excursion, and Deployment.

1.1.5.1 TYPES OF MACHINE LEARNING ARCHITECTURE

- **Supervised Learning**

In supervised learning, the training data used for is a mathematical model that consists of both inputs and desired outputs. Each corresponding input has an assigned output which is also known as a supervisory signal. Through the available training matrix, the system is able to determine the relationship between the input and output and employ the same in subsequent inputs post-training to determine the corresponding output. The supervised learning can further be broadened into classification and regression analysis based on the output criteria. Classification analysis is presented when the outputs are restricted in nature and limited to a set of values. However, regression analysis defines a numerical range of values for the output. Examples of supervised learning are seen in face detection, speaker verification systems.

- **Unsupervised Learning**

Unlike supervised learning, unsupervised learning uses training data that does not contain output. The unsupervised learning identifies relation input based on trends, commonalities, and the output is determined on the basis of the presence/absence of such trends in the user input.

- **Reinforcement Training**

This is used in training the system to decide on a particular relevance context using various algorithms to determine the correct approach in the context of the present state. These are widely used in training gaming portals to work on user inputs accordingly.

1.1.5.2 ARCHITECTING THE MACHINE LEARNING PROCESS

➢ Data Acquisition

As machine learning is based on available data for the system to make a decision hence the first step defined in the architecture is data acquisition. This involves data collection, preparing and segregating the case scenarios based on certain features involved with the decision making cycle and forwarding the data to the processingunit for carrying out further categorization. This stage is sometimes called the data preprocessing stage. The data model expects reliable, fast and elastic data which may be discrete or continuous in nature. The data is then passed into stream processing systems (for continuous data) and stored in batch data warehouses (for discrete data) before being passed on to data modeling or processing stages.

➢ Data Processing

The received data in the data acquisition layer is then sent forward to the dataprocessing layer where it is subjected to advanced integration and processing and involves normalization of the data, data cleaning, transformation, and encoding. The data processing is also dependent on the type of learning being used. For e.g., if supervised learning is being used the data shall be needed to be segregated into multiple steps of sample data required for training of the system and the data thus created is called training sample data or simply training data. Also, the data processing is dependent upon the kind of processing required and may involve choices ranging from action upon continuous data which will involve the use of specific function-based architecture, for example, lambda architecture, Also it might involve action upon discrete data which may require memory-bound processing. The data processing layer defines if the memory processing shall be done to data in transit or in rest.

➢ Data Modeling

This stage in machine learning is where the experimentation is done, testing is

involved and tunings are performed. The general goal behind being to optimize the algorithm in order to extract the required machine outcome and maximize the system performance, The output of the step is a refined solution capable of providing the required data for the machine to make decisions.

➢ Deployment

Operationalizing machine learning outputs refers to the process of deploying the outputs into a production environment, where they can be used to make real-time decisions. The outputs of a machine learning model are non-deterministic queries that need to be further processed in order to be useful in a decision-making system. Seamless integration of the machine learning output into the decision-making system can reduce the need for additional exploratory processing. This direct integration allows the machine to make decisions based on the output in real-time, which can improve efficiency and accuracy. The operationalization process helps to ensure that the machine learning output is easy to implement and maintain in a production environment. Proper operationalization can also reduce the risk of errors and improve the overall reliability of the system. The machine learning output should be thoroughly tested before deployment to ensure that it meets the desired specifications. The operationalization process should take into account the specific requirements of the decision-making system and should be optimized for performance and scalability. In conclusion, the operationalization of machine learning outputs is an important step in ensuring that the outputs are useful and effective in a real-world production environment.

2 SYSTEM STUDY

2.1 INTRODUCTION

This stage involves studying the existing system and interacting with the users, which determine user requirements and their expectation of the proposed system. Cost of incorporating changes required by the user is very less at this stage that steeply increases as development advance. System study is general term that refers to an orderly, structured process for identifying and solving problems. The first phase of software development is system study. The importance of system study phase is the establishment of the requirements for the system to be acquired, developed and installed. Analyzing the project to understand the complexity forms the vital part of the system study. Problematic areas are identified and information is collected. Fact finding or gathering is essential to any analysis of requirements. It is also highly essential that the analyst familiarize himself with the objectives, activities and functions of organizations in which the system is to be implemented.

System study works with users to identify goals and build systems to achieve them. System study is an important phase of any system development process. The system is studied to the minute details and analyzed. The system analyst plays the role of an interrogator and dwells deep into the working of the present system. In system study, a detailed study of these operations performed by a system and their relationship within and outside the system is done. A key question considered here is, “What must be done to solve the problem? One aspect of system study is defining the boundaries of the application and determining whether or not the candidate application should be considered.

The system is used as a whole and the inputs to the system are identified. The outputs from the system are traced various processing. During system study, data are collected on available sources, network data centers handled by the present system. Once system study is completed, the analyst has a firm understanding of what is to be done. If the information gathered in the system study is not enough to make a good software application, then we have to reschedules a new system study.

2.2 Feasibility Study

A feasibility analysis evaluates the candidate systems and determines the best system that needs performance requirements. The purpose of feasibility study is to investigate the present system, evaluate the possible application of computer-based methods, select a tentative system, evaluate the cost and effectiveness of the proposed system, evaluate the impact of proposed system on existing personnel and as certain the need for new personnel. Feasibility is carried out to see if the system is technically, economical and operationally feasible.

All projects are feasible when given unlimited resources and infinite time. It is both necessary and student to evaluate the feasibility of the project at the earliest possible time. A feasibility study is to warranted for systems in which economic justification is obvious technical risk is low, few legal problems are expected and no reasonable alternative exists. An estimate is made of whether the identified user may be satisfied using current hardware and software technologies. The study will decide if the proposed system will be cost effective from the business point of view and if it can be developed in the existing budgetary constraints. The result should inform the decision of whether to go ahead with a more detailed analysis.

2.2.2 Economic Feasibility

Economic analysis is the most commonly used method for evaluating effectiveness of a system. Cost-benefit analysis is the most important assessment of economic justification of the project. Cost-benefit analysis delineates the cost for project development and weighs them against tangible and intangible benefits of a system. This type of analysis varies with the characteristics of the system to be developed, the relative site of the project, and the expected return on investment. Benefits of a newsystem are always determined relative to the existing mode of operation. Economic feasibility deals about the economic impact faced by the organization to implement the new system. Not only cost of hardware,

software etc is considered but also the form of reduced costs. The project, installed certainly be beneficial since there will be reduction in manual work, increase in speed of work.

The analysis raises financial and economic questions during the preliminary investigation to estimate the following:

- The cost to conduct a full systems investigation.
- The cost of hardware and software for the class of application of the project beingconsidered.

To be judged feasible, a proposal for the specific project must pass all these tests, otherwise it is not considered as a feasible project. I gathered the details regarding the financial aspects incorporated in the system to make it costefficient.

2.2.2 Behavioral Feasibility

Proposed projects are beneficial only if they can be turned into information systems that will meet the operating requirements of the organization. This test of feasibility asks if the system will work when it is developed satisfies all the operational conditions. It was the most difficult task for me, but met efficiently.

As this package is found to be feasible technically, economically and functionally, the system is judged feasible. Viewing the collected information, recommendation and justification, conclusions is made of the proposed system. Hence decision is taken to go on with the project.

2.2.3 Technical Feasibility

There are a number of technical issues, which are generally raised during the feasibility stage of the investigation. A study of function, performance and constraints gave me the ability to achieve acceptable system. The software required for this system is:

- PYTHON
- OPENCV

2.2.4 Operational Feasibility

Suppose for a moment that technical and economic resources are both judged adequate. The systems analyst still considers the operational feasibility of the requested project. Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will operate and be used once it is installed. If users are virtually wed to the present system, see no problems with it, and generally are not involved inrequesting a new system, resistance to implementing the new system will be strong. Chances for it ever becoming operational are low.

2.2.5 Legal Feasibility

A determination of any in fragment, violation, or liability that could result from development of the system. Legal feasibility encompasses a broad range of concerns that include contracts, liability, infringement, and myriad other traps frequently unknown to technical staff.

2.3 EXISTING SYSTEM

The current attendance system in use relies on manual processes, such as taking attendance with pen and paper or using a roll call system. These methods are prone to errors, such as incorrect recording of attendance, and can be time-consuming. The manual data entry and tracking can lead to discrepancies in attendance records, affecting the accuracy of attendance reports. Additionally, the manual system may not be able to provide real-time updates or digital attendance records, making it difficult to track attendance and monitor compliance with attendance policies. These limitations have motivated the development of a more efficient and accurate attendance management system.

2.4 PROPOSED SYSTEM

As a solution to overcome the drawback of the existing system, the proposed system has been evolved. The proposed system aims to automate the attendance

process by utilizing facial recognition technology. The system will be trained on a dataset of facial images and will be able to accurately detect and match faces in real-time. By integrating with existing attendance management systems, the proposed system will provide attendance records in a digital format, improving the efficiency and accuracy of attendance taking. The use of OpenCV and Python will enable the creation of a robust and scalable system that can easily be integrated into existing systems. The proposed system will help organizations and educational institutions save time and resources, and ensure accurate attendance records. The end goal is to enhance and upgrade the current attendance system into a more efficient and effective solution.

2.5 Techniques for System Study

The most important source information is the end users of the current system, who are often also the potential users of the new system. They may range from novices to highly- skilled individuals. The information gathered from end users will be crucial during the analysis and design phases. Later, the analyst will also discuss technical aspects of the system with programmers, network engineers and other technical staff.

A secondary source of information for the analyst can be found in the existing paper work or document within the organization. Documents represent the formal information flow through the current system. The analyst must collect sample copies of all relevant documents, e.g. input forms, output document, report, invoices etc to understand how data flows and is used in the current system. This information can be important in the subsequent design of files for the new system. This analyst will use a range of techniques to gather information about the current system. The most important are interviewing, questioners and observation. In this project I have used observation as a technique for gathering information. Observation is a technique used to gather information by observing people performing various aspects of their loans. It allows the analyst to determine what is being done, how it is being done, who does it, when it is done, how long it takes, where it is done and why it is done.

3 SYSTEM ANALYSIS

3.1 INTRODUCTION

Once the scope of the study has been agreed, a detailed investigation is undertaken to determine the operations carried out by the current system and the requirements for the new system. This will involve speaking with the users of the current systems and examining all the paperwork involved. Analysis provides the foundation for the remainder of the systems analysis and design cycle. During the analysis phase, the analyst investigates task in the analysis phase is thegathering and analysis of information about the current system.

3.2 Hardware Specification

PROCESSOR	: PENTIUM 4
HARD DISK	: 20GB
CACHE MEMORY	: 8 M B
RAM	: 4 G B
MONITOR	: L C D MONITOR
KEYBOARD	: STANDARD KEYBOARD
MOUSE	: OPTICAL
CAMERA	: 5MP

3.3 Software Specification

OPERATING SYSTEM	: WINDOWS 10
CLASSIFIER USED	: HAAR-CASCADE CLASSIFIER
PROGRAMMING LANGUAGE	: PYTHON
COMPUTER VISION	: OPENCV
TERMINAL	: ANACONDA PROMPT

4 SYSTEM DESIGN

4.1 INTRODUCTION

We use LBPH and Machine Learning for Real Time Detection and Recognition of Human Faces, which is simple face detection and recognition system is proposed in this paper which has the capability to recognize human faces in single as well as multiple face images in a database in real time. Pre- processing of the proposed frame work includes noise removal ,hole filling and grayscaling of colour images. After pre-processing, face detection is performed by using LBPH architecture. Architecture layers of Classifier are created using numphy Library in Python. Detected faces are augmented to make computation fast. By using Local Binary Patterns Histograms algorithm features are extracted from the augmented image.

4.2 The Input Image

Real-time input images are used in this proposed system. Face of the person in input images must be fully visible without any distraction. The system requires a reasonable number of pixels and an acceptable amount of brightness for processing the image. Based on experimental evidence, it is supposed to perform well indoors as well as outdoors i.e. school ,college ,hospitals, police stations ,government office etc.

4.3 The Pre-processing Stage

The input image dataset needs to be loaded into Python data structures for pre-processing in order to remove any noise disturbances, enhance relevant features, and prepare the data for further analysis by the trained model. The pre-processing steps are necessary for the effective application of face detection and matching techniques. The pre-processing process consists of several steps, including noise removal, grayscaling, and hole filling. These steps help eliminate false detections of faces, and ensure that only true faces are detected.

After the pre-processing, the face image is then cropped and re-localized, which helps to focus the analysis on the most relevant parts of the image. Finally, histogram normalization is performed to improve the quality of the preprocessed image. This step helps to balance the color and brightness levels across the image, making it easier to detect and match faces accurately. In summary, the pre-processing of the input image is crucial for the effective application of face detection and matching techniques.

4.4 The Face Detection Stage

We perform face detection using HAAR Cascade algorithm.This system consists of the value of all black pixels in greyscale images was accumulated. They then deducted from the total number of white boxes. Finally, the outcome is compared to the given threshold, and if the criterion is met, the function considers it a hit.In general, for each computation in Haar-feature, each single pixel in the feature areas can need to be obtained, and this step can be avoided by using integral images in which the value of each pixel is equal to the number of grey values above and left in the image.

$$\text{Feature} = \Sigma ie\{1..N\} wi.\text{RecSum}(x, y, w, h),$$

where RecSum (x, y, w,h) is the summation of intensity in any given upright or rotated rectangle enclosed in a detection window and x, y,w,h is for coordinates, dimensions, and rotation of that rectangle, respectively. Haar Wavelets represented as box classifier which is used to extract face features by using integral image

4.5 The Feature-Extraction Stage

Feature extraction is a crucial step in improving the accuracy of a face recognition model. It involves the process of identifying and extracting important features from pre-processed face images. The extracted features are then transformed into a lower-dimensional representation, which retains the key characteristics of the original image. This stage of the process helps to simplify the data and reduces the computational burden of the model, making it more efficient. The reduced-dimensional representation is used to classify human faces, which is an essential task in face recognition. The feature extraction process is optimized to ensure that important facial features are preserved, while irrelevant information is discarded. This helps to improve the accuracy and robustness of the face recognition model.

4.6 The Training Stage

The face recognition method is based on Local Binary Patterns Histograms (LBPH) which is a machine learning model used for image classification. It is trained on preprocessed face images and uses the learned information to classify new images into different groups. The LBPH model performs two key stages in the face recognition process: feature extraction and classification. Feature extraction involves reducing the dimensionality of the face images and extracting important features that are used for classification.

The algorithm creates a histogram of the local binary patterns (LBP) of each pixel in an

image, which provides a unique feature set for each face. This feature set is then compared to the feature sets of faces stored in the training data.The selected features are then used to train the model and improve its accuracy. The trained model is saved and used in the prediction phase, where it is used to classify new faces . The use of LBPH, combined with the feature extraction and selection techniques, significantly improves the accuracy and efficiency of the face recognition system. This allows the model to accurately classify face images, even when faced with variations in lighting, pose, and facial expression.

4.1 The Recognition Stage

The LBPH Face Recognition method is used for recognition of faces. It is a machine learning algorithm used to identify and classify faces in an image or video. The algorithm creates a histogram of the local binary patterns (LBP) of each pixel in an image, which provides a unique feature set for each face. This feature set is then compared to the feature sets of faces stored in the training data. The algorithm calculates the similarity between the detected face and the faces in the training data by measuring the distance between their features. If the distance between the features is below a certain threshold, the face is recognized and identified. The threshold is set to ensure that the recognition is accurate and minimizes false positive results. The LBPH Face Recognition algorithm can be trained with a large dataset of faces to increase the accuracy of face recognition. The algorithm is computationally efficient and can be implemented on a variety of platforms, including embedded systems and real-time video streaming applications. The LBPH Face Recognition method is widely used in security systems, facial recognition systems, and other applications that require accurate face recognition.

4.2 Attendance marking Stage

Attendance is marked by updating the attendance record for a particular student. The algorithm starts by capturing video frames and detecting faces in the frame using the Haar cascade classifier. If a face is detected, it is then passed through the LBPH Face Recognizer to recognize the face. The algorithm then compares the features of the detected face with the features of the faces in the training data. If the face is recognized, the corresponding student's name is retrieved from the training data and used to mark attendance. This is typically done by updating a spreadsheet with the student's name and the date and time of the attendance mark. This process is repeated for every frame captured by the video until the video stream stops, at which point the attendance records for all students are updated and saved.

4.3 Automatic-Emailing Stage

The code is for automating the process of sending an email with an attached file. The yagmail library is used to handle the email sending process. The recipient's email address is stored in the receiver variable, the email body text is stored in the body variable, and the file to be attached is stored in the filename variable. The email credentials (username and password) are stored in the yagmail. SMTP method, which sets up the email account to be used for sending the email. The email is sent with the yag.send method, which takes the recipient's email address, email subject, email body, and the attached file as its arguments. Once the code is executed, the recipient will receive an email with the subject "Attendance Report" and the body text "Attendance File", along with the attached file

4.4 MODULES

The proposed system contains the following modules:

- Capture image ( )
- Training image ()
- Recognize (image)
- Auto mail

➢ Capture image ( )

The purpose of this module is to capture real-time images and data of students. This information can be used for training machine learning models. The Haar Cascade Classifier Model is utilized to capture the images, which is a popular machine learning algorithm used in computer vision and image processing.The input images are obtained from a webcam or camera in real-time, ensuring that the information gathered is up-to-date and accurate. The frames (images) undergo pre-processing, which includes cropping, resizing, and converting to grayscale, to prepare the images for further analysis. This step is crucial to ensure that the images are clean and free of any distractions that could impact the analysis.The processed images are saved to a designated directory, while the data collected about the students is stored in a CSV file. This provides a convenient and efficient way to access and store the information for future reference. The images and data can be used together to provide a more complete picture of the students.

➢ Training image ()

The training image process involves using computer vision techniques to train a machine learning model to recognize faces in images. The process starts by gathering a dateset of images of faces along with their corresponding labels. This dataset is stored in a directory.Then the images and labels from this directory are extracted and store them in two separate lists.

A face recognizer is created using the OpenCV's implementation of the Local Binary Pattern Histogram (LBPH). A haar cascade classifier is also initialized to detect faces in the images. The data extracted from the dataset are used to train the recognizer. This training process is done in a separate thread to allow for parallel processing.After the training process is complete, the trained model is saved to a file in a directory.

➢ Recognize (image)

The code uses OpenCV's LBPH (Local Binary Patterns Histograms) face recognition algorithm to recognize faces in real-time video captured by a webcam. The faces are detected using a Haar Cascade Classifier, which is a machine learning-based approach to detect objects in an image or video

The program reads a trained model Trainner.yml file, which contains facial recognition data for previously trained faces. Then it starts capturing video using the webcam, and detects faces using the Cascade Classifier. If the confidence level of a detected face is greater than 67%, it predicts the identity of the person using the LBPH Face Recognizer. If the confidence level is less than 67%, the person is considered "Unknown."

If the identity of a recognized person is found in the StudentDetails.csv file, it logs the attendance of that person in a pandas DataFrame with columns 'Id', 'Name', 'Date', and 'Time'. The detected DataFrame is then saved to a CSV file with a timestamped filename and sent as an email attachment using the automail module.

➢ Auto mail (image)

This code uses the yagmail library to send an email. First, it creates an SMTP object yag with the sender's email address and password. Then, it defines the receiver email address, the email subject, the email body, and the filename of the attachment. Finally, it uses the yag.send() method to send the email with the attachment to the receiver

4.5 DATA FLOW DIAGRAM

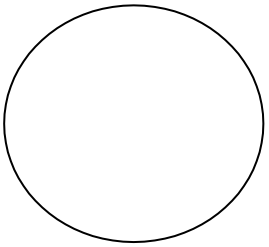
A DFD, also known as a “bubble chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. A DFD consists of a series of bubbles joined by lines. The bubbles represent data transformations and the lines represent data flow in the system. A data flow diagram may be used to represent a system or software at any level of abstraction. A DFD is a diagram that describes the flow of data and the processes that change or transform data throughout a system. It is a structured analysis and design tool that can be used or flowcharting in place of or in association with, information oriented and process oriented system flowchart. When analyst prepares the DFD, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply a physical implementation. The DFD reviews the current physical system, prepare input and output specification, specifies the implementation plan etc.

Basic data flow diagrams symbols are:

- A “Rectangle” defines a External Entity or source or destination of a system data

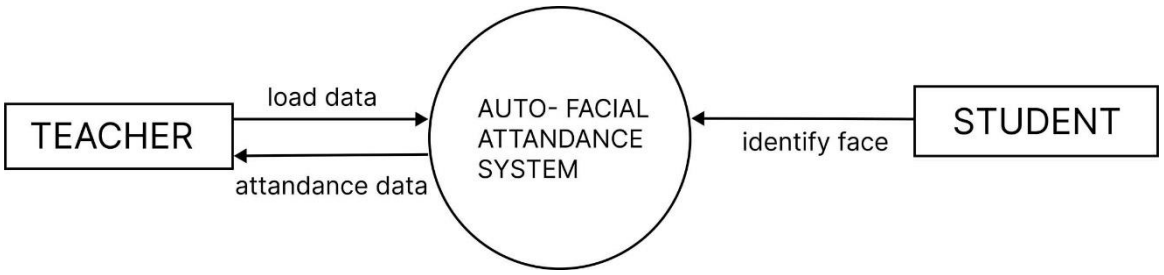
- An “Arrow” identifies data flow. It is a pipeline through which information flowS

- A “circle” represents a process that transforms incoming data flow(s) into outgoing dataflow(s).

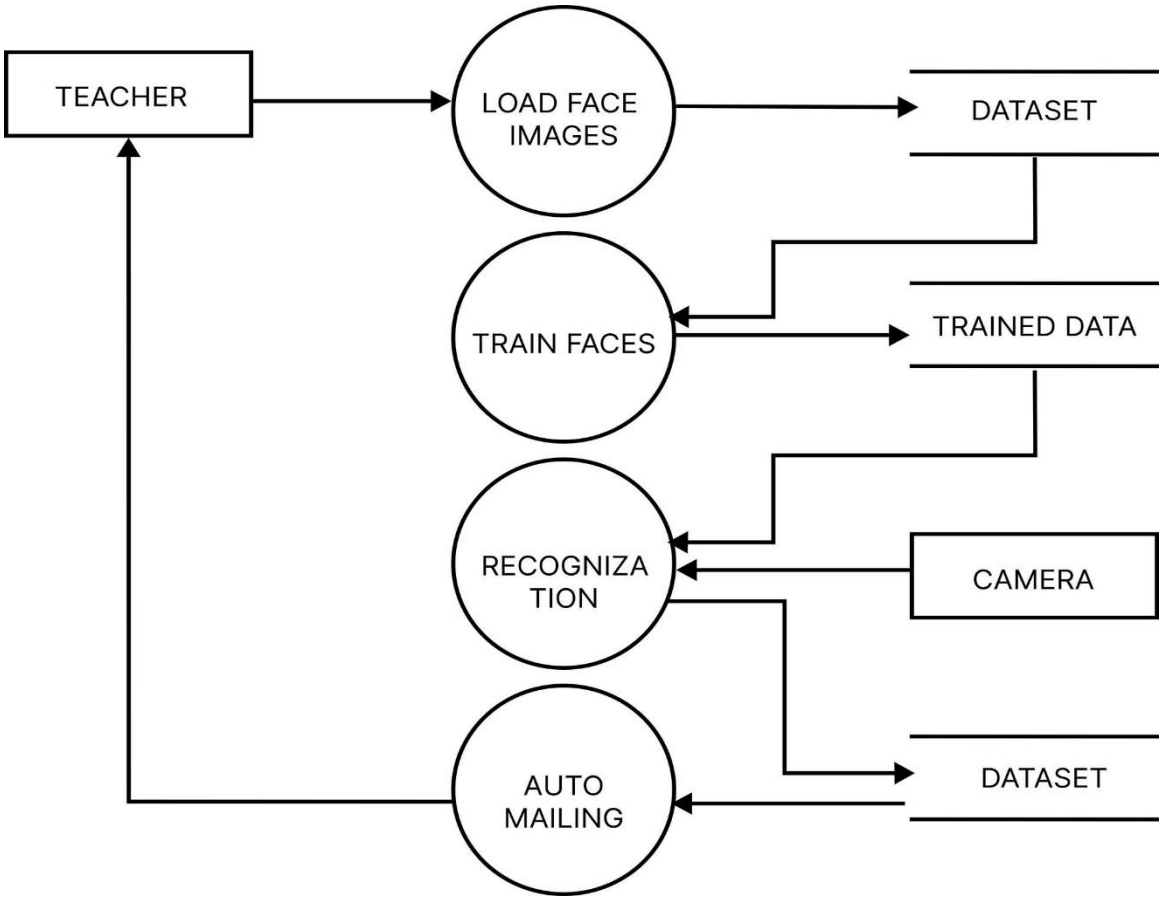


- An “Open rectangle” is a data store.

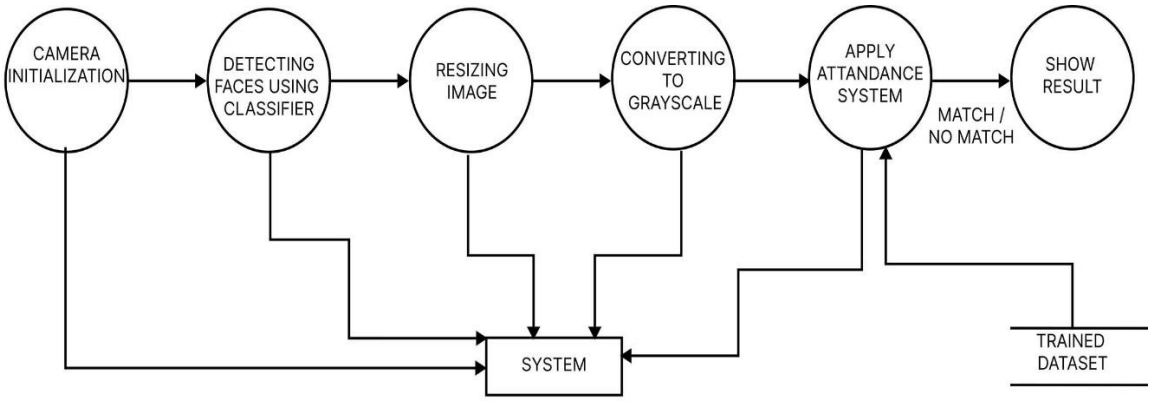
Level 0: Context Diagram



Level 1:



Level 2:



4.6 USE CASE DIAGRAM

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.

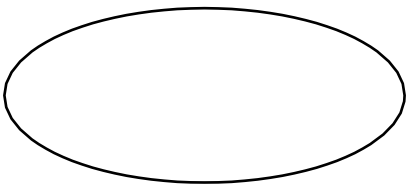
Basic Use-case Diagram Symbols are:

- **System**

  Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.

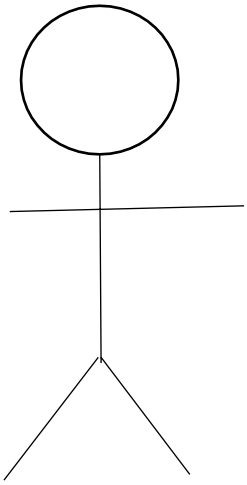
- **Use Case**

  Draw use cases using ovals. Label the ovals with verbs that represent the system's functions.



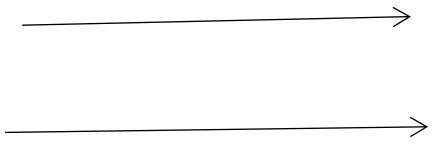
- **Actors**

  Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.

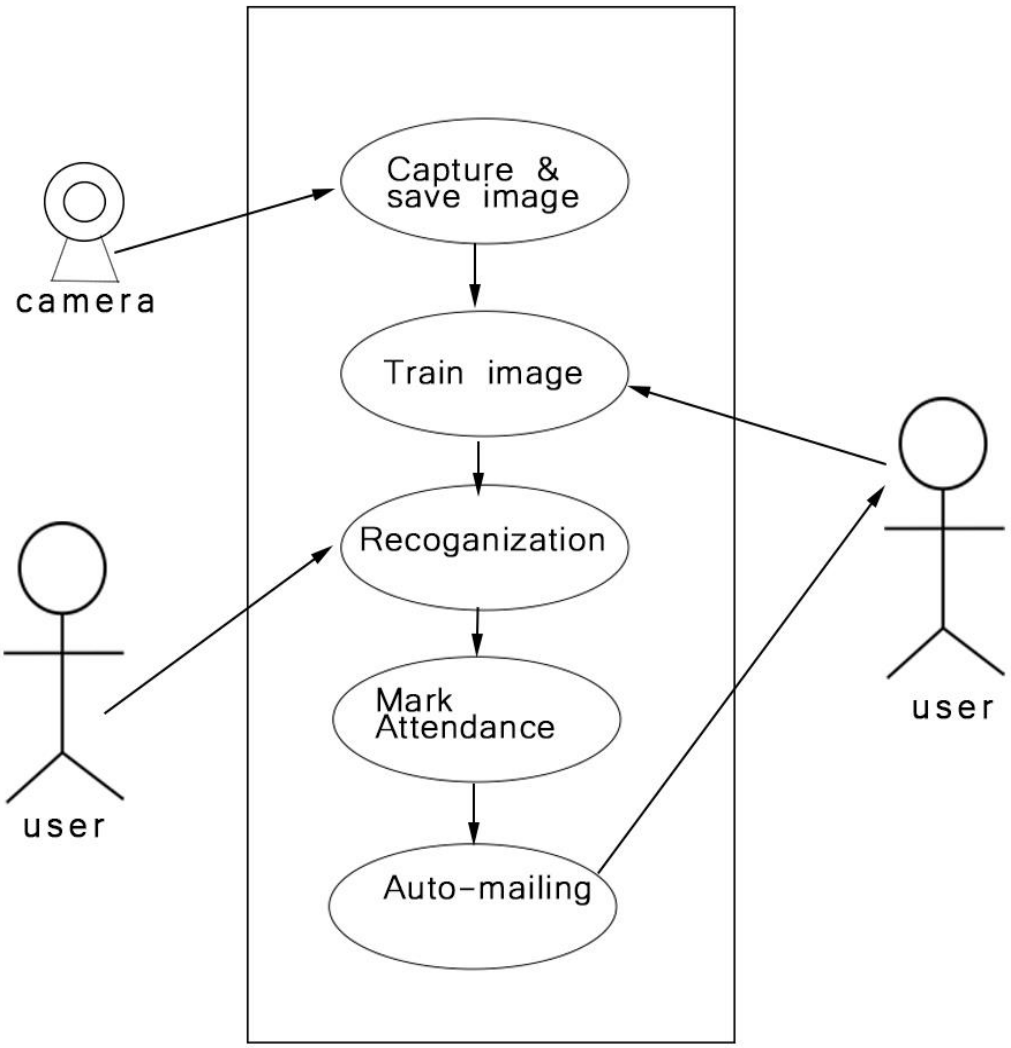


- **Relationships**

  Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labeled either "uses" or "extends." A "uses" relationship indicates that one use case is needed by another in order to perform a task. An "extends" relationship indicates alternative options under a certain use case.



DIAGRAM


Capture &
save image
camera
Train image
Recoganization
Mark
Attendance
Auto-mailing
user
user

5 SYSTEM TESTING

5.1 INTRODUCTION

Testing is an important phase in software development. After completion, the system may work without any problem. But, there should be several unknown or hidden errors in the system still remaining. The error chances may injected into the system at any stage of the development. Even if there are techniques to detect and eliminate the errors, some errors may retain in the system. So after the completion of coding, the system is to be executed with the only purpose of detecting maximum number of errors. The tester executes the system, and inputs different types of values those may cause error or some exceptional situation in the system. The error locations detected through the testing are to be corrected in the system then. So, the important and the only aim of testing is to detect and cure even a less possible of an error that may facein the future executions of the system. Testing is a set of activity that can be planned inadvance and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is completed without testing, as it is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are corrected, the goal will besuccessfully achieved. Inadequate testing or non-testing may lead to errors that may not appear until months later.

5.2 Purpose of Testing

Testing is the success of the system. System testing makes a logical assumption that if all part of the system is correct, the goal will be successfully achieved. The following points shows how testing is essential. Existence of program defects of inadequacies is inferred. Verifies whether the software behave as intended by its designer. Checks conformance with requirements specification or user need. Access the operational reliability of the system. Test the performance of the system. The performance of the system reflects the frequencies of actual user inputs. Find the fault which caused the output anomaly. Detect flaws and deficiencies in requirements. Exercise the program using data like the real data processed by the program. Test the system capabilities. Judges whether or not the program is usable in practice. Testing objectives there are several rules that can serve as testing objectives. They are; Testing is a process of executing a program with the intent of finding error.

A good test case is one that has high probability of finding an undiscovered error. A successful test is one that uncovers an undiscovered error.If testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrates that software functions appear to the working according to the specifications, that performance requirement appear to have been met. These are three ways to testa program For correctness For implementation efficiency For computational complexity. Test for correctness are supported to verify that a program does exactly what it was designed to do. This is much difficult that it may at first appear especially for large programs. Tests for implementation efficiency attempt to find ways to make a correct program faster or use less storage. It is a code-refining process, which reexamines the implementation phase algorithm development. Tests for computational complexity amount to an experiment analysis of the complexity of an algorithm or an experiment comparison of two or more algorithms, which solve the same problem.

5.3 Types of Testing

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct. The goal will be successfully achieves. The candidate system is subject to a variety of tests. A series of tests are performed for the proposed system is ready for system acceptance testing.

The various levels at which testing are conducted are:

- Unit testing
- Integration testing
- Sequential testing
- System testing
- Validation testing

Unit Testing

In unit testing each program unit is tested individually. so any errors in a unit are debugged. Sample data is given for unit testing. The unit test results are recorded for future references. Unit testing focus verification efforts on the smallest unit of software design, the module. This is known as “module testing”. It comprises of the set test performed by an individual programmer prior to the integration of unit into the large system. The modules are tested separately, this testing is carried out programming stage itself.

In this step each module is found to be working satisfactory as regard to the expected out from module. The unit testing was done for every module in the software for various inputs, such they each line of code is at least once executed. This testing was carried out during the unit to a large system.

Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting test to uncover errors associated with interfacing.

Program Testing

Program testing checks for two types of errors; syntax and logic. A syntax error is a program statement that violates one or more rules of the language in which it is written. A logic error deals with incorrect data fields. When a program is tested, the actual output is compared with the expected output. All the modules are combined and tested as a whole. Here correction is difficult because the vast expenses of all errors uncovered are correct for the next testing steps. We follow bottom-up integration. Bottom up integration testing as its name implies begin construction andsling with atomic modules. Because components are integrated from the bottom up, accessing required for the components subordinate to a given level is always available and need for stubs is eliminated.

Sequential Testing

Sequential or series testing is checking the logic of one or more programs in the candidate system, where the output of one program will affect the processing done by another program.

System Testing

System testing executing a program to check logic changes made in it and with the intension of finding errors-making the program fails. Effective testing does not guaranties reliability is a design consideration. This testing actually consists of a series of different test whose primary purpose is to fully exercise the computer based system. It begins where integration testing is completed and finally software is completely assembled as package ,interfacing errors are uncovered and corrected.

Acceptance Testing

Acceptance testing is running the system with live data by the actual user. An acceptance test has the objective of selling the user in the validity and reliability of the system. A comprehensive test report is prepared. The report indicates the system's tolerance, performance range, error rate and accuracy. It verifies the system procedures operate to system specification and the integrity of important data is maintained, performance of an acceptance test is actually the users show. User motivation is very important for the successful performanceof the system. After that a comprehensive test report is prepared. This report shows the systems tolerance, performance range, error rate and accuracy.

Input Testing

System testing involves testing the system with all possible combinations of inputs, including user-entered data such as passwords and numerical details. This helps to identify any defects or issues in the system's handling of inputs. The system should respond with appropriate error messages in case of invalid or incorrect inputs. This type of testing ensures the system's user-friendliness and helps to improve the overall user experience. Testing with all verifiable combinations of inputs is a critical aspect of system testing to ensure the quality and reliability of the system.

Output Testing

Here the output is tested to view where the screen is what which is desired. It is also checked whether it is to the satisfaction of the user. Changes that need to be done can be done after the result is seen

5.4 System Implementation

A crucial phase in the system life cycle is the successful implementation of the new system design. Implementation involves creating computer compatible files, training the operating staff, installing hardware, terminals. In the system implementation, user training is crucial for minimizing resistance to change and giving the new system a chance to prove its worth. The objectives of the system implementation is to put the system into operation while holding costs, risks and personal irritation to minimum. Once the physical system has been designed in details, the next stage is to run the design into a working systemand then to monitor the operation of the system to ensure that is continue to work efficiently and the operation of the system to ensure that is continues to work efficiently and effectively. The implementation stage of a is often very complex and time consuming because many more people are involved than in the earlier stages.

The system implementation took place through various stages as follows,

- Implantation planning.
- Education and training.
- System testing.
- System implementation.
- Change over.

The implementation plan includes a description of all the activities that must occur to implement the new system and to put it into operation. To achieve the objectives and benefits from computer based system, it is essential for the people who will be confident of their role in the new jobs. After software is developed to meet user's requirements, users test it for acceptance. The change over phase is used to provide adaptability for the new system.

5.5 System Maintenance

Software maintenance is the process of modifying a software system or component after its delivery in order to correct faults,improve the performance and other attributes ,or to adapt to the changed environment .maintenance covers a wide range of activities including correcting the error and design coding, updating the documentation and test data , and upgrading the user hardware and software .maintenance is always necessary to keep the software usable and useful. Hardware also requires periodic maintenance to keep the system into its standards. After installation is completed and user start is adjust to the changes created by the candidate system.evaluation and maintenance begin. If new information is consistent with design specification the changes have to be made. Hardware also requires periodic maintenance to keep in tune with design specifications. User priorities changes in organizational requirements or environmental factors also called for system enhancements. Maintenance covers wide range of activities, including correcting, coding and design errors, updating documentation and test data ,andupgrading user support.any activities classified as maintenance are actually enhancements.

Maintenance means restoring something to do its original condition. Unlike hardware, software does not wear outfit is corrected. In contrast, enhancement means adding, modifying or redeveloping the code to support changes in the specifications. It is necessary to keep up with changing user needs the operational environment. Maintenance means repairing processing or performance failures or making changes because of previously uncorrected problems or false assumptions .adaptive maintenance means changing the program function. Perfective maintenance means enhancing the performance or modifying the program to respond to the user's additional or changing needsof these types more time and money are spend on perfective than on corrective and adaptive maintenance together. Maintenance activities begin where conversion leaves off. Maintenance is handled by the same planning and control used in a formal system project .a major problem with Software maintenance is its labor intensive nature. documentation is as much a part of

maintenance as it is of system development .to put maintenance in its proper perspective requires considerable skill and experience and is an important and is an important and ongoing aspect of system development is an additional factor in the success of the maintenance programmer is the work environment.

SOFTWARE MAINTENANCE ACTIVITIES CAN BE CLASSIFIED INTO:

- Corrective maintenance.
- Adaptive maintenance.
- Perceptive maintenance.

Corrective maintenance removes software faults. Perfective maintenance improves the system without changing its functionality.the objective of perfective maintenance should be to prevent failures and optimize the software.Adaptive maintenance modifies the software to keep it up to date with its operative environment.it may be needed because of changes in the user requirements ,changes in target platform, or changes in external interfaces. Minor adaptive changes should be handled by normal maintenance process major adaptive changes should be carried out as, a separate development project.If you want to changes the software to improve future maintainability orreliability or to provide better basis for future enhancement then perceptive maintenance is perform.

6 CONCLUTION

In conclusion, the development of an automatic facial attendance system using OpenCV and Python is a significant step in utilizing technology to improve the attendance process in organizations and educational institutions. The system's facial recognition technology will allow for efficient and accurate attendance taking, reducing ambiguity and improving accuracy. The use of OpenCV and Python provides a robust and scalable solution that can easily be integrated into existing systems. The project's main objective is to enhance and upgrade the traditional attendance system, making it more efficient and effective.

The use of face recognition technology as a unique identifier of individuals ensures a low deviation or duplication rate, making it a reliable solution for attendance taking. The system's ability to store attendance information digitally and integrate it into existing systems will provide a more organized and streamlined process. The creation of a face database to pump data into the recognizer algorithm and the automatic marking of attendance will save time and resources compared to manual attendance taking. The end of the day excel sheet, containing attendance information, can be mailed to the faculty, providing them with a quick and easy access to attendance records.

In conclusion, the project to develop an automatic facial attendance system using OpenCV and Python offers a solution to improve the efficiency and accuracy of the attendance process in educational institutions. The combination of facial recognition technology and the use of OpenCV and Python provides a robust and scalable solution that can be easily integrated into existing systems. The project is a significant step forward in utilizing technology to simplify and improve essential processes within educational institutions.

7 FUTURE ENHANCEMENTS

FUTURE ENHANCEMENT

The automatic facial attendance system developed using OpenCV and Python has the potential for further improvement and enhancement in the future. One area that can be improved is the accuracy of the system in recognizing and matching faces. This can be achieved by training the system using advanced deep learning algorithms and larger facial image datasets. Another enhancement can be the integration of multi-modal verification, combining facial recognition with other biometric technologies such as fingerprint or iris recognition to increase the reliability of the system. The system can also be integrated with real-time monitoring systems to provide real-time updates on attendance and alert administrators in case of any discrepancies. Additionally, the system can be integrated with Learning Management Systems to provide real-time attendance updates to students and teachers and provide a more comprehensive solution for educational institutions. A mobile app can also be developed for students to check their attendance, providing a convenient solution for remote or online learning environments. Finally, the system can be moved to a cloud-based infrastructure, providing scalability and accessibility from anywhere with an internet connection.

8 BIBLOGRAPHY

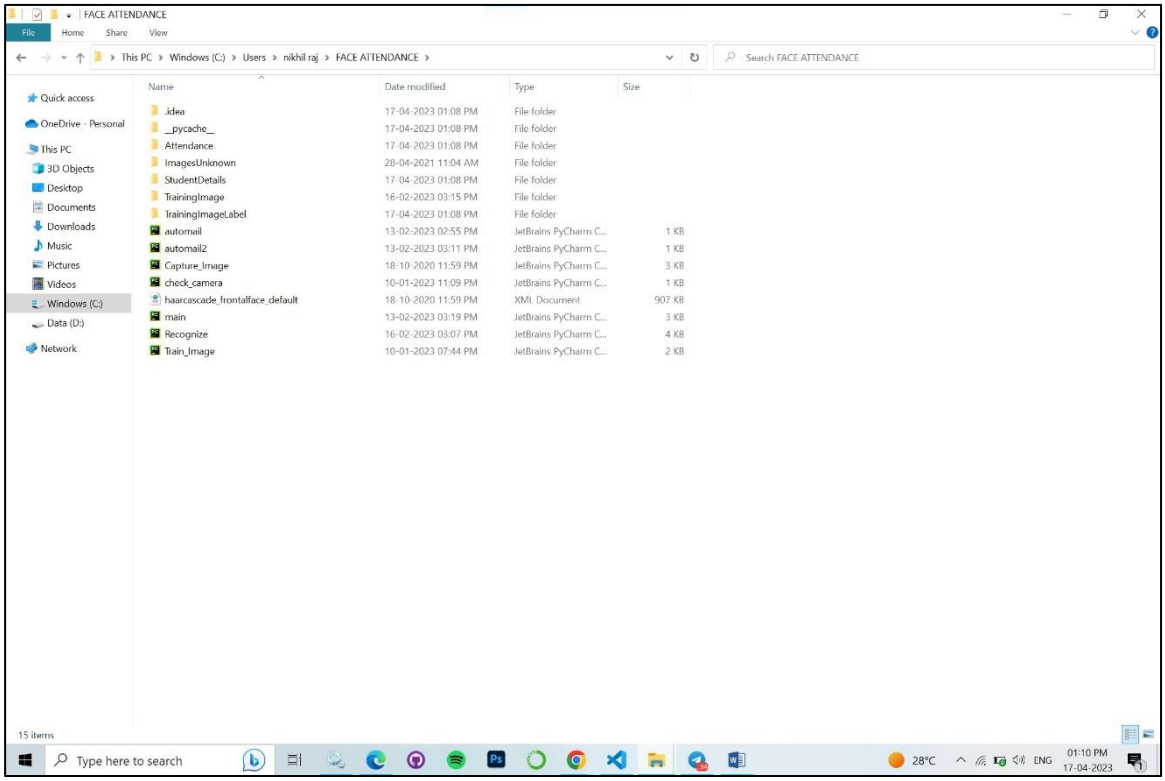
8.1 Text Book Reference

- Zheng Jun, Hua Jizhao, Tang Zhenglan, Wang Feng “Face detection based on LBP",2017 IEEE 13th International Conference on Electronic Measurement & Instruments
- Q. B. Sun, W. M. Huang, and J. K. Wu "Face DetectionBased on Color and Local Symmetry Information”, National University of Singapore Heng Mui Keng Terrace, Kent Ridge Singapore.
- Based on Color and Local Symmetry Information”, National University of Singapore Heng Mui Keng Terrace, Kent Ridge Singapore
- Based on Color and Local Symmetry Information”, National University of Singapore Heng Mui Keng Terrace, Kent Ridge Singapore
- Based on Color and Local Symmetry Information”, National University of Singapore Heng Mui Keng Terrace, Kent Ridge Singapore
- T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," IEEE Transactions on pattern analysis and machine intelligence, vol. 23, pp. 681-685, 2001.
- T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," IEEE Transactions on pattern analysis and machine intelligence, vol. 23, pp. 681-685, 2001.
- S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017.
- Matthew D Zeiler, Rob Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV 2014: Computer Vision – ECCV 2014 pp 818-833.
- Matthew D Zeiler, Rob Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV 2014: Computer Vision – ECCV 2014 pp 818-833.

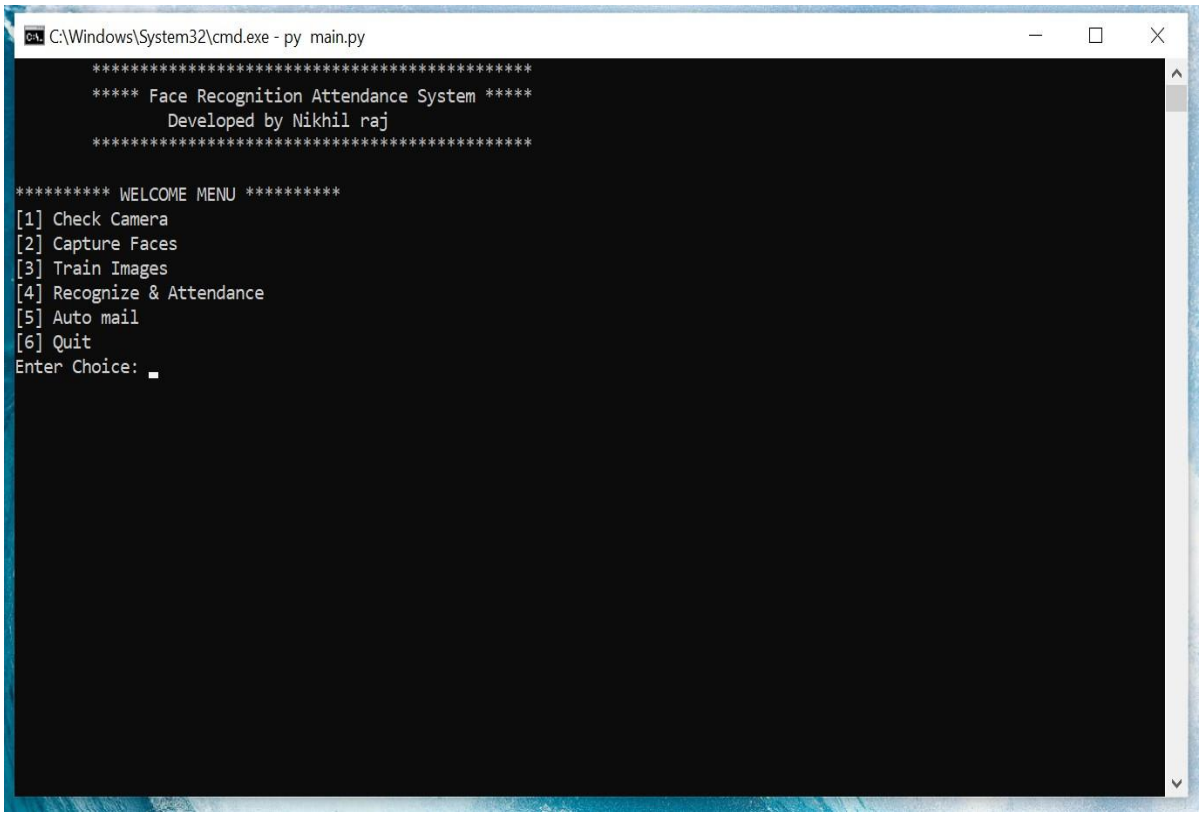
9 APPENDIX

9.1 SCREENSHOT

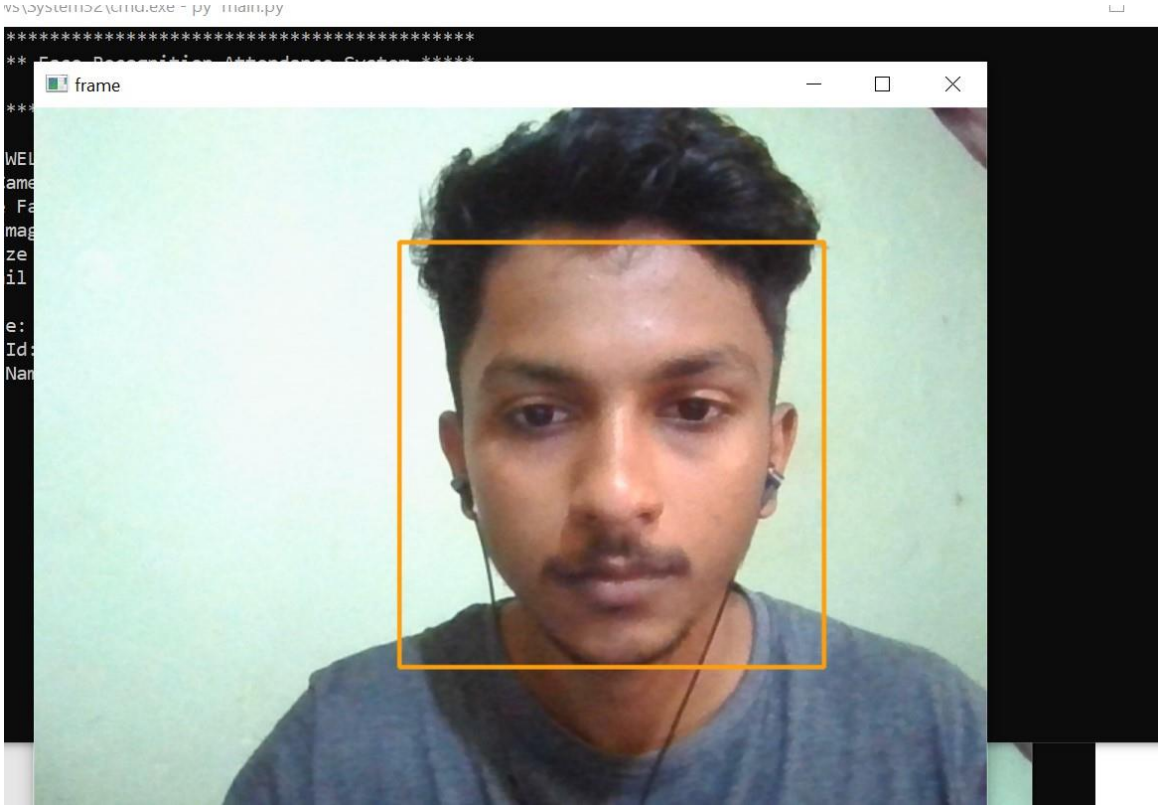
PATH:



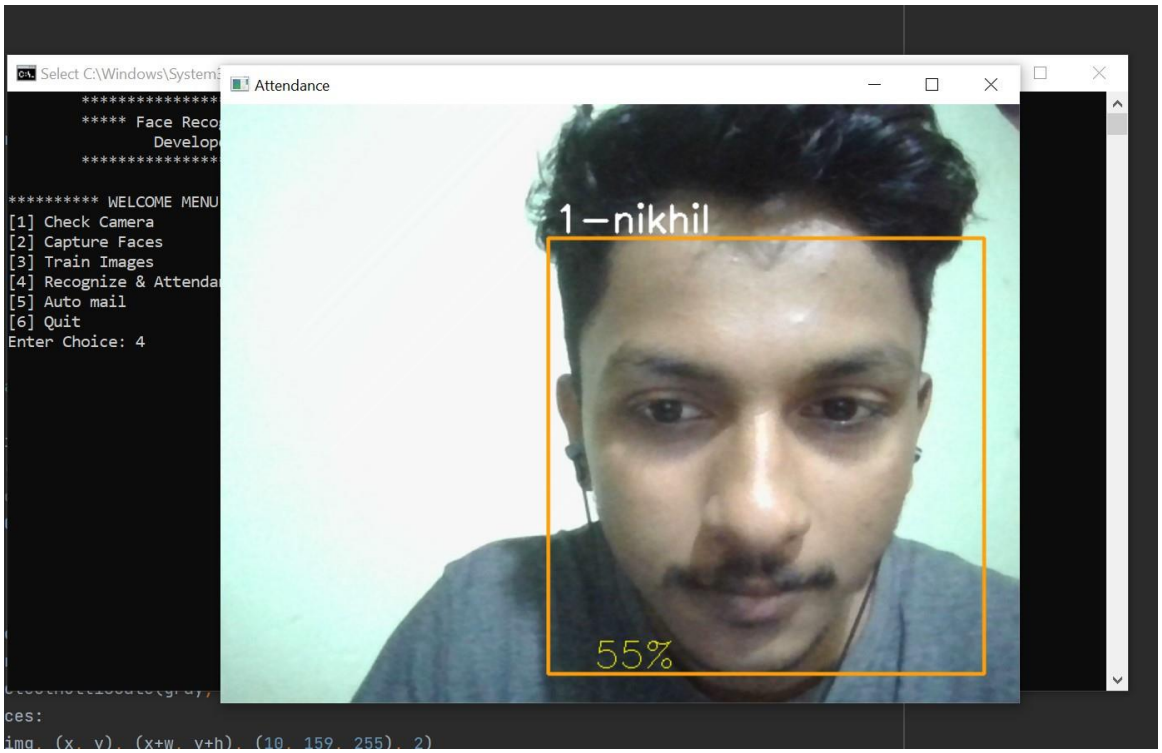
USER INTERFACE



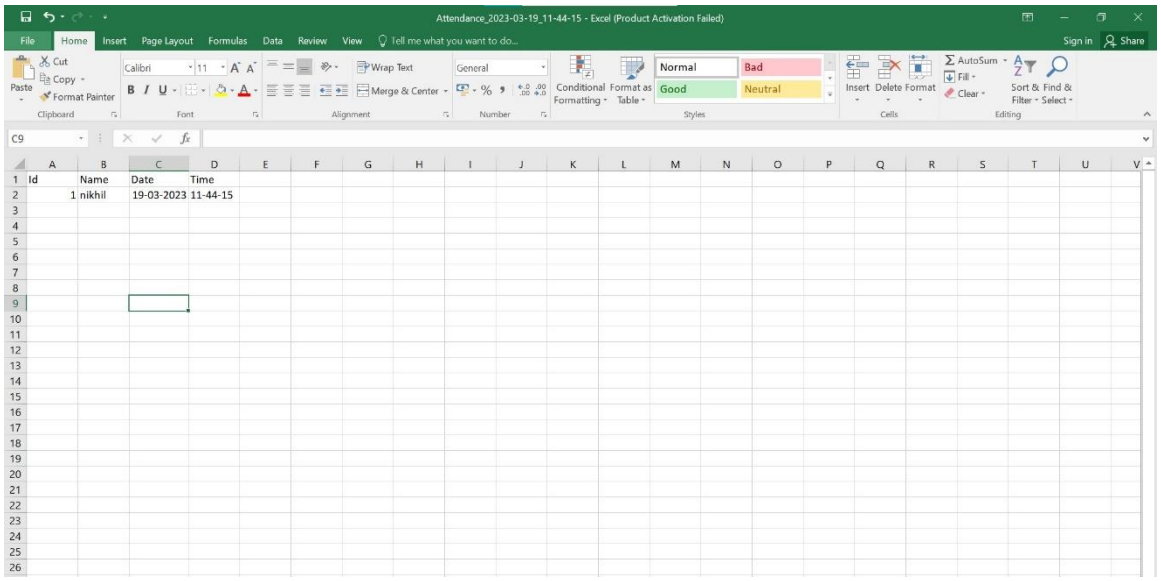
LIVE CAPTURED IMAGES:



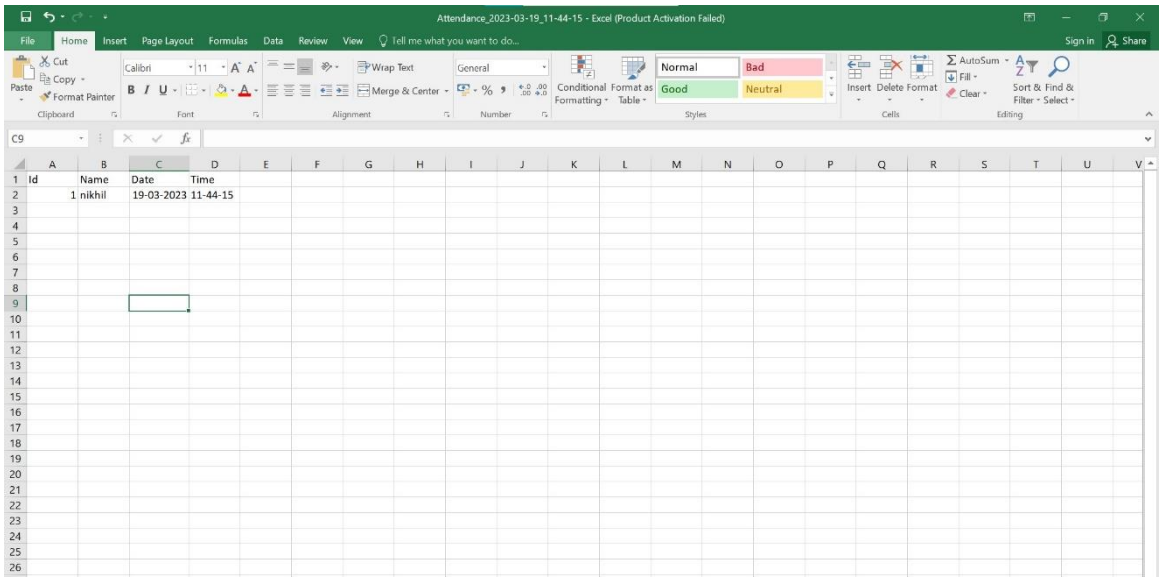
DETECTING FACE:



STUDENT DETAILS:



ATTANDANCE DETAILS:



9.2 Sample Codes

Main.py

```
import os
import check_camera
import Capture_Image
import Train_Image
import Recognize

def title_bar():
    os.system('cls')  # for windows

    print("\t**********************************************")
    print("\t***** Face Recognition Attendance System *****")
    print("\t       Developed by Nikhil raj              ")
    print("\t**********************************************")

def mainMenu():
    title_bar()
    print()
    print(10 * "*", "WELCOME MENU", 10 * "*")
    print("[1] Check Camera")
    print("[2] Capture Faces")
    print("[3] Train Images")
    print("[4] Recognize & Attendance")
    print("[5] Auto mail")
    print("[6] Quit")

    while True:
        try:
            choice = int(input("Enter Choice: "))

            if choice == 1:
                checkCamera()
                break
            elif choice == 2:
                CaptureFaces()
                break
            elif choice == 3:
                Trainimages()
                break
            elif choice == 4:
                RecognizeFaces()
                break
            elif choice == 5:
                os.system("py automail2.py")
                break
                mainMenu()
            elif choice == 6:
```

```
                print("Thank You for using attendance management system :) ")
                break

            else:
                print("Invalid Choice. Enter 1-4")
                mainMenu()
        except ValueError:
            print("Invalid Choice. Enter 1-4\n Try Again")
    exit

def checkCamera():
    check_camera.camer()
    key = input("Enter any key to return main menu")
    mainMenu()

def CaptureFaces():
    Capture_Image.takeImages()
    key = input("Enter any key to return main menu")
    mainMenu()

def Trainimages():
    Train_Image.TrainImages()
    key = input("Enter any key to return main menu")
    mainMenu()

def RecognizeFaces():
    Recognize.recognize_attendence()
    key = input("Enter any key to return main menu")
    mainMenu()
mainMenu()
```

Capture_image.py

```
import csv
import cv2
import os

def is_number(s):
    try:
        float(s)
        return True
    except ValueError:
        pass

    try:
        import unicodedata
        unicodedata.numeric(s)
        return True
    except (TypeError, ValueError):
        pass
```

```
    return False


def takeImages():


    Id = input("Enter Your Id: ")
    name = input("Enter Your Name: ")

    if(is_number(Id) and name.isalpha()):
        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, minSize=(30,30),flags =
cv2.CASCADE_SCALE_IMAGE)
            for(x,y,w,h) in faces:
                cv2.rectangle(img, (x, y), (x+w, y+h), (10, 159, 255), 2)
                #incrementing sample number
                sampleNum = sampleNum+1
                #saving the captured face in the dataset folder TrainingImage
                cv2.imwrite("TrainingImage" + os.sep +name + "."+Id + '.' +
                            str(sampleNum) + ".jpg", gray[y:y+h, x:x+w])
                #display the frame
                cv2.imshow('frame', img)
            #wait for 100 miliseconds
            if cv2.waitKey(100) & 0xFF == ord('q'):
                break
            # break if the sample number is more than 100
            elif sampleNum > 100:
                break


        cam.release()
        cv2.destroyAllWindows()
        res = "Images Saved for ID : " + Id + " Name : " + name
        row = [Id, name]
        with open("StudentDetails"+os.sep+"StudentDetails.csv", 'a+') as csvFile:
            writer = csv.writer(csvFile)
            writer.writerow(row)
        csvFile.close()
    else:
        if(is_number(Id)):
            print("Enter Alphabetical Name")
        if(name.isalpha()):
            print("Enter Numeric ID")
```