# Introduction

Attendance maintenance is a significant function in all the institutions to monitor the performance of the students. Every institute does this in its own way. Some of these institutes use the old paper or file-based systems and some have adopted strategies of automatic attendance using some biometric techniques. A facial recognition system is computerized biometric software which is suited for determining or validating a person by performing comparison on patterns based on their facial appearances. Face recognition systems have upgraded appreciably in their management over the recent years and this technology is now vastly used for various objectives like security and in commercial operations. Face recognition is a powerful field of research which is a computer based digital technology. Face recognition for the intent of marking attendance is a resourceful application of attendance system. It is widely used in security systems and it can be compared with other biometrics such as fingerprint or eye iris recognition systems. As the number of students in an educational institute or employees at an organization increases, the needs for lecturers or to the organization also increase the complication of attendance control. This project may be helpful for the explanation of these types of problems. The number of students present in a lecture hall is observed, each person is identified and then the information about the number of students who are present I maintained.

Face recognition is a non-invasive identification system and faster than other systems since multiple faces can be analyzed at the same time. The difference between face detection and identification is, face detection is to identify a face from an image and locate the face. Face recognition is making the decision “whose face is it?”, using an image database. In this project both are accomplished using different techniques and are described below. The report begins with a brief history of face recognition. This is followed by the explanation of HAAR-cascades, Eigenface, Fisherface and Local binary pattern histogram (LBPH) algorithms. Next, the methodology and the results of the project are described. A discussion regarding the challenges and the resolutions are described. Finally, a conclusion is provided on the pros and cons of each algorithm and possible implementations.

## **1.1 Problem statement**

The concept of face recognition is to give a computer system the ability of finding and recognizing human faces fast and precisely in images or videos. Numerous algorithms and techniques have been developed for improving the performance of face recognition. Recently Deep learning has been highly explored for computer vision applications. Human brain can automatically and instantly detect and recognize multiple faces. But when it comes to computer, it is very difficult to do all the challenging tasks on the level of human brain. The face recognition is an integral part of biometrics. In biometrics, basic traits of human are matched to the existing data. Facial features are extracted and implemented through algorithms, which are efficient and some modifications are done to improve the existing algorithm models. Computers that detect and recognize faces could be applied to a wide variety of practical applications including criminal identification, security systems, and identity verification like Real-time face verification system on a cell-phone using advanced correlation filters.

## **1.2 Aims and Objectives**

Improve and organize the process of track and manage student attendance and absenteeism. Additionally, we seek to:

* Provides a valuable attendance service for both teachers and students.
* Reduce manual process errors by provide automated and a reliable attendance system uses face recognition technology.
* Increase privacy and security which student cannot presenting himself or his friend while they are not.
* Produce monthly reports for lecturers.
* Flexibility, Lecture’s capability of editing attendance records.
* Calculate absenteeism percentage and send reminder messages to students.

## **1.3 Python**

[Python](https://www.geeksforgeeks.org/python-programming-language/) is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**Characteristics of Python**

Following are important characteristics of Python Programming −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* It supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

**Usage of Python:-**

Python is a general purpose, open source, high-level programming language and also provides number of libraries and frameworks. Python has gained popularity because of its simplicity, easy syntax and user-friendly environment. The usage of Python as follows.

* Desktop Applications
* Web Applications
* [Data Science](https://www.javatpoint.com/data-science)
* [Artificial Intelligence](https://www.javatpoint.com/artificial-intelligence-tutorial)
* [Machine Learning](https://www.javatpoint.com/machine-learning)
* Scientific Computing
* [Robotics](https://www.javatpoint.com/robotics-tutorial)
* [Internet of Things (IoT)](https://www.javatpoint.com/iot-internet-of-things)
* Gaming
* Mobile Apps
* Data Analysis and Preprocessing

**Why Python?**

Python is easy to use, powerful, and versatile, making it a great choice for beginners and experts alike. Python’s readability makes it a great first programming language — it allows you to think like a programmer and not waste time with confusing syntax.

**1.4 Tkinter**

**Tkinter** is the most commonly used library for developing GUI (Graphical User Interface) in Python. It is a standard Python interface to the Tk GUI toolkit shipped with Python. As Tk and Tkinter are available on most of the Unix platforms as well as on the Windows system, developing GUI applications with Tkinter becomes the fastest and easiest.The name Tkinter comes from Tk interface. Tkinter was written by Fredrik Lundh. Tkinter is the inbuilt python module that is used to create GUI applications. It is one of the most commonly used modules for creating GUI applications in Python as it is simple and easy to work with. You don’t need to worry about the installation of the Tkinter module separately as it comes with Python already. It gives an object-oriented interface to the Tk GUI toolkit.

**Python Tkinter:-**

Python provides the standard library Tkinter for creating the graphical user interface for desktop based applications.

Developing desktop based applications with python Tkinter is not a complex task. An empty Tkinter top-level window can be created by using the following steps.

1. import the Tkinter module.
2. Create the main application window.
3. Add the widgets like labels, buttons, frames, etc. to the window.
4. Call the main event loop so that the actions can take place on the user's computer screen.

There are a number of widgets which you can put in your tkinter application. Some of the major widgets are explained below:

|  |  |
| --- | --- |
| **Sr.No.** | **Operator & Description** |
| 1 | [**Button**](https://www.tutorialspoint.com/python/tk_button.htm)  The Button widget is used to display buttons in your application. |
| 2 | [**Canvas**](https://www.tutorialspoint.com/python/tk_canvas.htm)  The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application |

|  |  |
| --- | --- |
|  |  |
| 3 | [**Checkbutton**](https://www.tutorialspoint.com/python/tk_checkbutton.htm)  The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time. |
| 4 | [**Entry**](https://www.tutorialspoint.com/python/tk_entry.htm)  The Entry widget is used to display a single-line text field for accepting values from a user. |
| 5 | [**Frame**](https://www.tutorialspoint.com/python/tk_frame.htm)  The Frame widget is used as a container widget to organize other widgets. |
| 6 | [**Label**](https://www.tutorialspoint.com/python/tk_label.htm)  The Label widget is used to provide a single-line caption for other widgets. It can also contain images. |
| 7 | [**Listbox**](https://www.tutorialspoint.com/python/tk_listbox.htm)  The Listbox widget is used to provide a list of options to a user. |
| 8 | [**Menubutton**](https://www.tutorialspoint.com/python/tk_menubutton.htm)  The Menubutton widget is used to display menus in your application. |
| 9 | [**Menu**](https://www.tutorialspoint.com/python/tk_menu.htm)  The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton. |
| 10 | [**Message**](https://www.tutorialspoint.com/python/tk_message.htm)  The Message widget is used to display multiline text fields for accepting values from a user. |
| 11 | [**Radiobutton**](https://www.tutorialspoint.com/python/tk_radiobutton.htm)  The Radiobutton widget is used to display a number of options as radio buttons. The user can select only one option at a time. |
| 12 | [**Scale**](https://www.tutorialspoint.com/python/tk_scale.htm) |

|  |  |
| --- | --- |
|  | The Scale widget is used to provide a slider widget. |
| 13 | [**Scrollbar**](https://www.tutorialspoint.com/python/tk_scrollbar.htm)  The Scrollbar widget is used to add scrolling capability to various widgets, such as list boxes. |
| 14 | [**Text**](https://www.tutorialspoint.com/python/tk_text.htm)  The Text widget is used to display text in multiple lines. |
| 15 | [**Toplevel**](https://www.tutorialspoint.com/python/tk_toplevel.htm)  The Toplevel widget is used to provide a separate window container. |
| 16 | [**Spinbox**](https://www.tutorialspoint.com/python/tk_spinbox.htm)  The Spinbox widget is a variant of the standard Tkinter Entry widget, which can be used to select from a fixed number of values. |
| 17 | [**PanedWindow**](https://www.tutorialspoint.com/python/tk_panedwindow.htm)  A PanedWindow is a container widget that may contain any number of panes, arranged horizontally or vertically. |
| 18 | [**LabelFrame**](https://www.tutorialspoint.com/python/tk_labelframe.htm)  A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts. |
| 19 | [**tkMessageBox**](https://www.tutorialspoint.com/python/tk_messagebox.htm)  This module is used to display message boxes in your applications. |

## 

## **1.5 OpenCV**

OpenCV (Open Source Computer Vision Library) is a [library of programming functions](https://en.wikipedia.org/wiki/Library_(computing)) mainly aimed at real-time [computer vision](https://en.wikipedia.org/wiki/Computer_vision). Originally developed by [Intel](https://en.wikipedia.org/wiki/Intel_Corporation), it was later supported by [Willow Garage](https://en.wikipedia.org/wiki/Willow_Garage) then Itseez (which was later acquired by Intel). The library is [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) and free for use under the [open-source](https://en.wikipedia.org/wiki/Open-source_software) [Apache 2 License](https://en.wikipedia.org/wiki/Apache_License). Starting with 2011, OpenCV features GPU acceleration for real-time operations.

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

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding [18 million](https://sourceforge.net/projects/opencvlibrary/files/stats/timeline?dates=2001-09-20+to+2019-01-30). The library is used extensively in companies, research groups and by governmental bodies.

**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.

**Why OpenCV?**

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today’s systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as Numpy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

**1.6 NumPy**

NumPy is a library for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)), adding support for large, multi-dimensional [arrays](https://en.wikipedia.org/wiki/Array_data_structure) and [matrices](https://en.wikipedia.org/wiki/Matrix_(math)), along with a large collection of [high-level](https://en.wikipedia.org/wiki/High-level_programming_language) [mathematical](https://en.wikipedia.org/wiki/Mathematics) [functions](https://en.wikipedia.org/wiki/Function_(mathematics)) to operate on these arrays.[[5]](https://en.wikipedia.org/wiki/NumPy#cite_note-Nature-5) The ancestor of NumPy, Numeric, was originally created by [Jim Hugunin](https://en.wikipedia.org/wiki/Jim_Hugunin) with contributions from several other developers. In 2005, [Travis Oliphant](https://en.wikipedia.org/wiki/Travis_Oliphant) created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is [open-source software](https://en.wikipedia.org/wiki/Open-source_software) and has many contributors.

NumPy targets the [CPython](https://en.wikipedia.org/wiki/CPython" \o "CPython) reference [implementation](https://en.wikipedia.org/wiki/Programming_language_implementation) of Python, which is a non-optimizing [bytecode](https://en.wikipedia.org/wiki/Bytecode) interpreter. Mathematical algorithms written for this version of Python often run much slower than [compiled](https://en.wikipedia.org/wiki/Compiler) equivalents. NumPy addresses the slowness problem partly by providing multidimensional arrays and functions and operators that operate efficiently on arrays, requiring rewriting some code, mostly [inner loops](https://en.wikipedia.org/wiki/Inner_loop), using NumPy.

Using NumPy in Python gives functionality comparable to [MATLAB](https://en.wikipedia.org/wiki/MATLAB) since they are both interpreted,[[18]](https://en.wikipedia.org/wiki/NumPy#cite_note-18) and they both allow the user to write fast programs as long as most operations work on arrays or matrices instead of [scalars](https://en.wikipedia.org/wiki/Scalar_(computing)). In comparison, MATLAB boasts a large number of additional toolboxes, notably [Simulink](https://en.wikipedia.org/wiki/Simulink), whereas NumPy is intrinsically integrated with Python, a more modern and complete programming language. Moreover, complementary Python packages are available; [SciPy](https://en.wikipedia.org/wiki/SciPy) is a library that adds more MATLAB-like functionality and [Matplotlib](https://en.wikipedia.org/wiki/Matplotlib) is a plotting package that provides MATLAB-like plotting functionality. Internally, both MATLAB and NumPy rely on [BLAS](https://en.wikipedia.org/wiki/Basic_Linear_Algebra_Subprograms) and [LAPACK](https://en.wikipedia.org/wiki/LAPACK) for efficient linear algebra computations.

**Why NumPy?**

NumPy is an open-source numerical Python library. NumPy contains a multi-dimensional array and matrix data structures. It can be utilized to perform a number of mathematical operations on arrays such as trigonometric, statistical, and algebraic routines. Therefore, the library contains a large number of mathematical, algebraic, and transformation functions. NumPy is an extension of Numeric and Numarray. Numpy also contains random number generators. NumPy is a wrapper around a library implemented in C. Pandas objects rely heavily on NumPy objects. Essentially, Pandas extends Numpy.

## **1.7 Scikit-Learn**

Scikit-learn (formerly scikits.learn and also known as sklearn) is a [free software](https://en.wikipedia.org/wiki/Free_software) [machine learning](https://en.wikipedia.org/wiki/Machine_learning) [library](https://en.wikipedia.org/wiki/Library_(computing)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) [programming language](https://en.wikipedia.org/wiki/Programming_language). It features various [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and [clustering](https://en.wikipedia.org/wiki/Cluster_analysis) algorithms including [support vector machines](https://en.wikipedia.org/wiki/Support_vector_machine), [random forests](https://en.wikipedia.org/wiki/Random_forests), [gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting), [k-means](https://en.wikipedia.org/wiki/K-means_clustering) and [DBSCAN](https://en.wikipedia.org/wiki/DBSCAN), and is designed to interoperate with the Python numerical and scientific libraries [NumPy](https://en.wikipedia.org/wiki/NumPy) and [SciPy](https://en.wikipedia.org/wiki/SciPy).

Scikit-learn is largely written in Python, and uses [numpy](https://en.wikipedia.org/wiki/Numpy" \o "Numpy) extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in [Cython](https://en.wikipedia.org/wiki/Cython" \o "Cython) to improve performance. Support vector machines are implemented by a Python wrapper around [LIBSVM](https://en.wikipedia.org/wiki/LIBSVM); logistic regression and linear support vector machines by a similar wrapper around [LIBLINEAR](https://en.wikipedia.org/wiki/LIBLINEAR). In such cases, extending these methods with Python may not be possible.

Scikit-learn integrates well with many other Python libraries, such as [matplotlib](https://en.wikipedia.org/wiki/Matplotlib" \o "Matplotlib) and [plotly](https://en.wikipedia.org/wiki/Plotly" \o "Plotly) for plotting, [numpy](https://en.wikipedia.org/wiki/NumPy" \o "NumPy) for array vectorization, [pandas](https://en.wikipedia.org/wiki/Pandas_(software)) dataframes, [scipy](https://en.wikipedia.org/wiki/SciPy" \o "SciPy), and many more.

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

The library is built upon the SciPy (Scientific Python) that must be installed before you can use scikit-learn. This stack that includes:

* **NumPy**: Base n-dimensional array package
* **SciPy**: Fundamental library for scientific computing
* **Matplotlib**: Comprehensive 2D/3D plotting
* **IPython**: Enhanced interactive console
* **Sympy**: Symbolic mathematics
* **Pandas**: Data structures and analysis

Extensions or modules for SciPy care conventionally named [SciKits](https://scikits.appspot.com/scikits). As such, the module provides learning algorithms and is named scikit-learn.

Features:

The library is focused on modeling data. It is not focused on loading, manipulating and summarizing data. For these features, refer to NumPy and Pandas.

Some popular groups of models provided by scikit-learn include:

* Clustering: for grouping unlabeled data such as KMeans.
* Cross Validation: for estimating the performance of supervised models on unseen data.
* Datasets: for test datasets and for generating datasets with specific properties for investigating model behavior.
* Dimensionality Reduction: for reducing the number of attributes in data for summarization, visualization and feature selection such as Principal component analysis.
* Ensemble methods: for combining the predictions of multiple supervised models.
* Feature extraction: for defining attributes in image and text data.
* Feature selection: for identifying meaningful attributes from which to create supervised models.
* Parameter Tuning: for getting the most out of supervised models.
* Manifold Learning: For summarizing and depicting complex multi-dimensional data.
* Supervised Models: a vast array not limited to generalize linear models, discriminate analysis, naive bayes, lazy methods, neural networks, support vector machines and decision trees.

# 

# 

# Software and H ardware Requirements

## **2.1 Hardware Requirement**

The following hardware components were used in developing the system: -

* + Processor- Core i3
  + RAM – 8GB
  + Storage- 80GB
  + CPU -2.99MHZ

**Running Platform**

The following hardware components are required in running the system: -

* + Processor - Pentium 3 processor
  + RAM- Minimum 128MB
  + HDD- Minimum 4.3GB
  + Keyboard- 103 keys

## **2.2 Software Requirement**

The following software components were used in developing the system: -

* + Operating System-Windows 10
  + Front End- Idle, Pycharm
  + Back End- Python
  + Editor – Pycharm, visual studio

##### **Running Platform**

The following software components are required in running the system: -

* + Operating System- Windows 2000/XP/7/8/10
  + Application Server - Jupyter Notebook
  + Front End – Idle, pycharm
  + Back End - Python

# Implementation

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 JavaScript 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. And we know that both the languages cannot communicate with each other directly so we have used IPC (Inter Personal Communication) techniques with zero libraries as a bridge to communicate these two languages. The Electron JS call the python functions and interchange data via TCP with help of Zero PC Library.

**A. Data Acquisition**

i. 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.

ii. 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.

iii. Storing: We have used JSON to store the student’s data.

iv. JavaScript Object Notation (JSON): To represent a structured data based on JavaScript object syntax, a standard text based format is introduced. JSON is used for transmitting data in web application. It is a perfect solution for storing temporary data that’s consumed by the entity that’s creates the data. JSON can store data in String, Number, Object, Array, Boolean, Null form which means it has limitation of storing data in functions, dates and undefined data form. If you are trying to store or exchange data in functions or dates than JSON is not right choice for you.

**B. Face recognition process**

**Figure 1. Face RECOGNITION PROCESS**

**i. 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 introducing 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.

**ii. 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.

**iii. 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.

**iv. 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.

**v. Attendance Marking:** Once the face is identifying with the image stored in JSON 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 creates separate worksheet by subject id, so that attendance is differentiated for different subjects.

## **3.1 Image processing**

The facial recognition process can be split into two major stages: processing which occurs before detection involving face detection and alignment and later recognition is done using feature extraction and matching steps.

**1. FACE DETECTION**

The primary function of this step is to conclude whether the human faces emerge in a given image, and what is the location of these faces. The expected outputs of this step are patches which contain each face in the input image. In order to get a more robust and easily designable face recognition system.

Face alignment is performed to rationalize the scales and orientation of these patches.

**2. FEATURE EXTRACTION**

Following the face detection step the extraction of human face patches from images is done. After this step, the conversion of face patch is done into vector with fixed coordinates or a set of landmark points.

**3. FACE RECOGNITION**

The last step after the representation of faces is to identify them. For automatic recognition we need to build a face database. Various images are taken foe each person and their features are extracted and stored in the database. Then when an input image is fed the face detection and feature extraction is performed and its feature to each face class is compared and stored in the database.

**4. ALGORITHMS**

There are various algorithms used for facial recognition.

Some of them are as follows:

i. Eigen faces

ii. Fisher faces

iii. Local binary patterns histograms

**i. Eigen faces:** This method is a statistical plan. The characteristic which influences the images is derived by this algorithm. The whole recognition method will depend on the training database that will be provided. The images from two different classes are not treated individually. Eigen face is based on PCA that classify images to extract features using a set of images. It is important that the images are in the same lighting condition and the eyes match in each image. Also, images used in this method must contain the same number of pixels and in grayscale. Each raw is concatenated to create a vector, resulting a 1×n 2 matrix. All the images in the dataset are stored in a single matrix resulting a matrix with columns corresponding the number of images. The matrix is averaged (normalized) to get an average human face. By subtracting the average face from each image vector unique features to each face are computed. In the resulting matrix, each column is a representation of the difference each face has to the average human face. The next step is computing the covariance matrix from the result. To obtain the Eigen vectors from the data, Eigen analysis is performed using principal component analysis. From the result, where covariance matrix is diagonal, where it has the highest variance is considered the 1st Eigen vector. 2nd Eigen vector is the direction of the next highest variance, and it is in 90 degrees to the 1st vector. 3rd will be the next highest variation, and so on. Each column is considered an image and visualized, resembles a face and called Eigen faces. When a face is required to be recognized, the image is imported, resized to match the same dimensions of the test data as mentioned above. By projecting extracted features on to each of the Eigen faces, weights can be calculated. These weights correspond to the similarity of the features extracted from the different image sets in the dataset to the features extracted from the input image. The input image can be identified as a face by comparing with the whole dataset. By comparing with each subset, the image can be identified as to which person it belongs to. By applying a threshold detection and identification can be controlled to eliminate false detection and recognition. PCA is sensitive to large numbers and assumes that the subspace is linear. If the same face is analyzed under different lighting conditions, it will mix the values when distribution is calculated and cannot be effectively classified. This makes to different lighting conditions poses a problem in matching the features as they can change dramatically.

**ii. FISHER FACES:**Fisher faces algorithm also follows a progressive approach just like the Eigen faces. This method is a alteration of Eigen faces so it uses the same principal Components Analysis. The major conversion is that the fisher faces considers the classes. As mentioned previously, the Eigen faces does not differentiate between the two pictures from two differed classes while training. The total average affects each picture. A Fisher face employs Linear Discriminate Analysis for distinguishing between pictures from a different class. Fisher face is one of the popular algorithms used in face recognition, and is widely believed to be superior to other techniques, such as eigenface because of the effort to maximize the separation between classes in the training process. The purpose of this research is to establish a program of face recognition application using fisher face method by utilizing GUI applications and databases that are used in the form of a Papuan facial image. Image recognition using fisher face method is based on the reduction of face space dimension using Principal Component Analysis (PCA) method, then apply Fisher's Linear Discriminate (FDL) method or also known as Linear Discriminate Analysis (LDA) method to obtain feature of image characteristic. The algorithm used in the process for image recognition is fisher faces algorithm while for identification or matching face image using minimum Euclidean. The method used in this study is literature study that is studying and reviewing various books or literature related to mathematical concepts that underlie the formation of fisher face algorithm to recognize the image of a person's face which is then applied in programming language, especially programming language Matlab7.10. While in the process of preprocessing used Adobe Photoshop CS4 application program, its goal is to make the face image to be uniform in terms of size and format so that the image is ready to be used by the system. The results show that for image recognition where the image of testing is the same as the training image, the percentage of program success is 100%, while for 73 facial test images with various expressions and various positions, 70 faces are recognized correctly and 3 faces are recognized incorrectly, so the percentage of success is 93%.

**iii. LOCAL BINARY PATTERNS HISTOGRAMS**: This method needs the gray scale pictures for dealing with the training part. This algorithm in comparison to other algorithms is not a holistic approach.

PARAMETERS:LBPH uses the following parameters:

Radius:Generally, 1 is set as a radius for the circular local binary pattern which denotes the radius around the central pixel.

Neighbors: The number of sample points surrounding the central pixel which is generally 8. The computational cost will increase with increase in number of sample points.

Grid X:The number of cells along the horizontal direction is represented as Grid X. With the increase in number of cells the grid becomes finer which results in increase of dimensional feature vector.

Grid Y**:** The number of cells along the vertical direction is represented as Grid Y. With the increase in number of cells the grid becomes finer which results in increase of dimensional feature vector.

**ALGORITHM TRAINING**

For the training purpose of the dataset of the facial images of the people to be recognized along with the unique ID is required so that the presented approach will utilize the provided information for perceiving an input image and providing the output. Same images require same ID.

**COMPUTATION OF THE ALGORITHM**

The intermediate image with improved facial characteristics which corresponds to the original image is created in the first step. Based on the parameters provided, sliding window theory is used in order to achieve so. Facial image is converted into gray scale. A 3x3 pixels window is taken which can also be expressed as a 3x3 matrix which contains the intensity of each pixel (0-255). After this we consider the central value of the matrix which we take as the threshold. This value defines the new values obtained from the 8 neighbors’. A new binary value is set for each neighbor of the central value. For the values equal to or greater than the threshold value 1 will be the output otherwise 0 will be the output. Only binary values will be present in the matrix and the concatenation is performed at each position to get new values at each position. Then the conversion of this binary value into a decimal value is done which is made the central value of the matrix. It is a pixel of the actual image. As the process is completed, we get a new image which serves as the better characteristics of the original image.

**EXTRACTION OF HISTOGRAM**

The image obtained in the previous step uses the Grid X and Grid Y parameters and the image is split into multiple grids. Based on the image the histogram can be extracted as below:

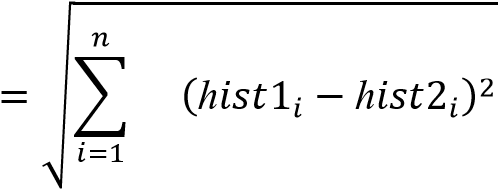
|  |  |  |
| --- | --- | --- |
| **FACE** | **FACE** |  |
| Confidence factor based on output is 2,000-3,000. | It is 100-400. |  |
| Threshold value is 4,000. | Threshold value is 400 | Threshold value is  7. |
| Principle of dataset generation is component based. | It is component based. | It is pixel based. |
| Basic principle is PCA. | Basic principle is  LDA. | Basic principle is Histogram. |
| Background noise is maximum. | Background noise is medium. | Background noise is minimum. |
| Efficiency is minimum. | Efficiency is greater than  Eigen face. | Efficiency is maximum. |

i. The image is in gray scale and each histogram will consist of only 256 positions (0-255) which symbolize the existences of each pixel intensity.

ii. After this each histogram is created and a new and bigger histogram is done. Let us suppose that there are 8x8 grids, and then there will be 16.384 positions in total in the final histogram. Ultimately the histogram signifies the features of the actual image.

**THE FACE RECOGNITION**

The training of the algorithm is done. For finding the image which is same as the input image, the two histograms are compared and the image corresponding to the nearest histogram is returned. Different approaches are used for the calculation of distance between the two histograms. Here we use the Euclidean distance based on the formula:

𝐷 

Hence the result of this method is the ID of the image which has the nearest histogram. It should return the distance calculated in the form of **‘confidence’**. Then the threshold and the ‘confidence’ can be used to automatically evaluate if the image is correctly recognized. If the confidence is less than the given threshold value, it implies that the image has been well recognized by the algorithm.

Table 1. Comparison of LBPH with other algorithms.

ADVANTAGES OF USING LBPH ALGORITHM*:*

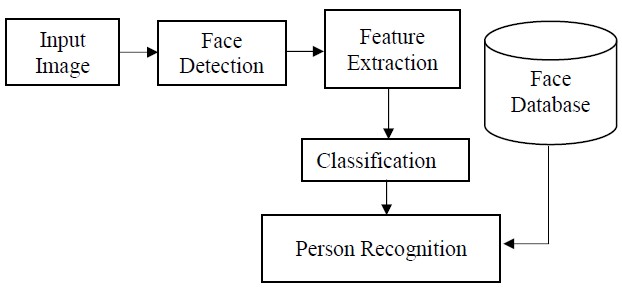
It is one of the simplest algorithms for face recognition.

The local features of the images can be characterized by this algorithm.

Using this algorithm, considerable results can be obtained.

Open CV library is used to implement LBPH algorithm.

## **3.2 Block diagram**



**Figure 2. Block Diagram**

**DATABASE CREATION:**

The first step in the Attendance System is the creation of a database of faces that will be used. Different individuals are considered and a camera is used for the detection of faces and the recording of the frontal face. The number of frames to be taken for consideration can be modified for accuracy levels. These images are then stored in the database along with the Registration ID.

**TRAINING OF FACES:**

The images are saved in gray scale after being recorded by a camera. The LBPH recognizer is employed to coach these faces because the coaching sets the resolution and therefore the recognized face resolutions are completely variant. A part of the image is taken as the centre and the neighbors’ are threshold against it. If the intensity of the centre part is greater or equal than it neighbor then it is denoted as 1 and 0 if not. This will result in binary patterns generally known as LBP codes.

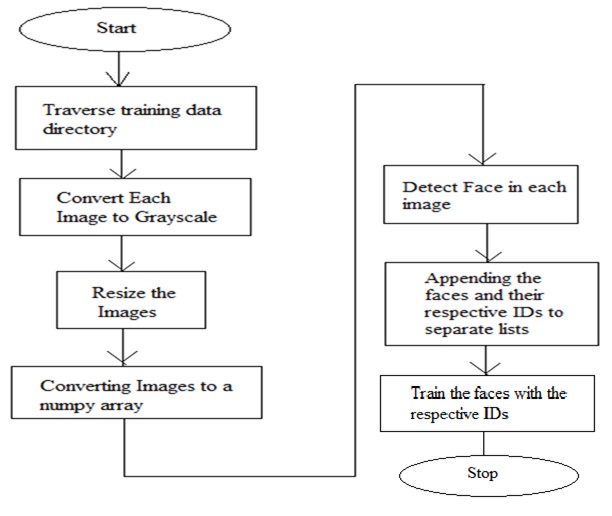
**FACE DETECTION**

The data of the trained faces is stored in .py format. The faces are detected using the Haar cascade frontal face module.

**FACE RECOGNITION***:*

The data of the trained faces are stored and the detected faces are compared to the IDs of the students and recognized. The recording of faces is done in real time to guarantee the accuracy of the system. This system is precisely dependant on the camera’s condition.

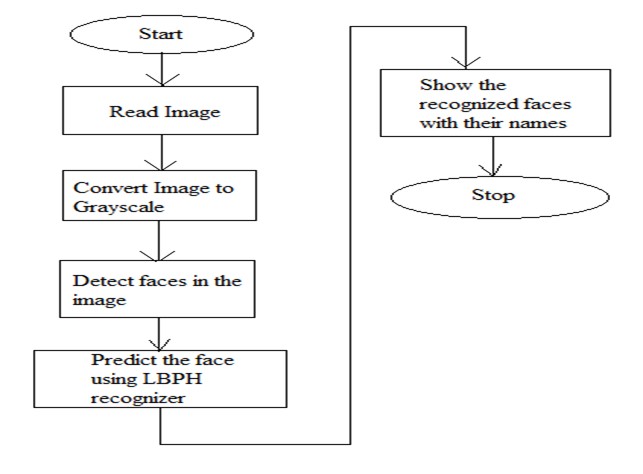
## **3.3 Flow chart**



**Figure 3. Flow-chart of the methodology used for Training Process**

The training process starts with traversing of the training data directory. Each image in the training date is converted into gray scale. A part of the image is taken as center and threshold its neighbours against it. If the intensity of the middle part is more or equal than its neighbor then denote it with 1 and 0 if not. After this the images are resized. Then the images are converted into a numpy array which is the central data structure of the numpy library. Each face in the image is detected. Creation of separate lists of each face is done and the faces are appended into them along with their respective IDs.

The faces are then trained with their respective IDs.



**Fig 3. Flow-chart of the methodology used for Face Detection and Recognition**

The input image is read by the camera of the phone. After the image is read it is converted into gray scale. The faces in the image are detected using the Haar Cascade frontal face module. Using the LBPH algorithm, the faces in the image are predicted. After the images are predicted, the recognized faces are shown in a green box along with their names.

## **3.4 Software description**

**OpenCV**

Open CV (Open Source Computer Vision Library) is a open source computer vision software library for the purpose of machine learning. Open CV was developed to serve the purpose of computer vision applications and to stimulate the usage of machine perception in the commercially viable products. Open CV is a BSD- licensed product which is easy for the utilization and modification of the code. The library contains more than 2500 advanced algorithms including an extensive set of both typical and state-of-the-art computer vision and machine learning algorithms. These algorithms can be employed for the detection and recognition of faces, identification of objects, extraction of 3 D models of objects, production of 3 D point clouds from stereo cameras, stitching images together for production of a high resolution image of an entire scene, finding similar images from an image database, removing red eyes from images taken using flash, following ye movements, recognition of scenery and establishing markers to overlay it with intensified reality etc. It includes C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. Open CV mainly involves real-time vision applications taking advantage of MMX and SSE instructions when available. A full-featured CUDA and Open CL interfaces are being progressively developed. There are over 500 algorithms and about 10 times functions that form or back those algorithms. Open CV is

written inherently in C++ and has a template interface that works harmoniously with STL containers.

**Pandas**

Pandas is an open source Python package that caters diverse tools for data analysis. The package contains various data structures that can be used for many diverse data manipulation tasks. It also includes a range of methods that can be invoked for data analysis, which becomes feasible when working on data science and machine learning problems in Python.

**Idle**

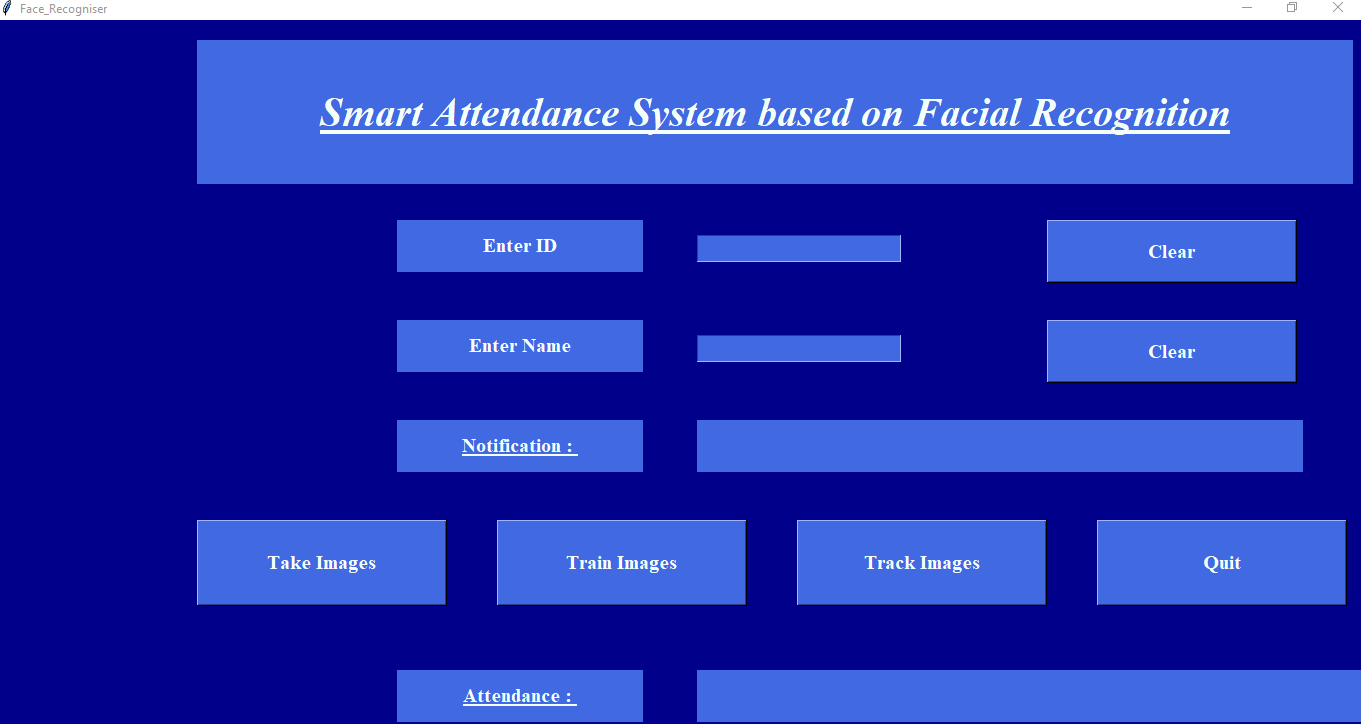
IDLE is Pythons Integrated Development and Learning Environment. IDLE is completely coded in Python, using the tkinter GUI toolkit. It works mostly uniformly on Windows, Unix and macOS. It has a Python shell window (interactive interpreter) with colorizing of error messages, code input and code output. There is a multi-window text editor with multiple undo, Python colorizing, smart indent, call tips, auto completion, and other features. Searching within any window, replacing within editor windows and searching through multiple files is possible. It also has configuration, browsers and other dialogs as well.

**Microsoft Excel**

Microsoft Excel is a spreadsheet program incorporated in Microsoft Office suite of applications. Spreadsheets prompt tables of values arranged in rows and columns that can be mathematically manipulated using both basic and complex arithmetic functions and operations. Apart from its standard spreadsheet features, Excel also extends programming support via Microsofts Visual Basic for Applications (VBA), the capacity to access data from external sources via Microsofts Dynamic Data Exchange (DDE) and extensive graphing and charting abilities. Excel being electronic spreadsheet program can be used to store, organize and manipulate the data. Electronic spreadsheet programs were formerly based on paper spreadsheets used for accounting purpose. The basic layout of computerized spreadsheets is more or less same as the paper ones. Related data can be stored in tables – which are a group of small rectangular boxes or cells that are standardized into rows and columns.

## **3.4 How the system works?**

This system works accordingly with a series of step explained below:

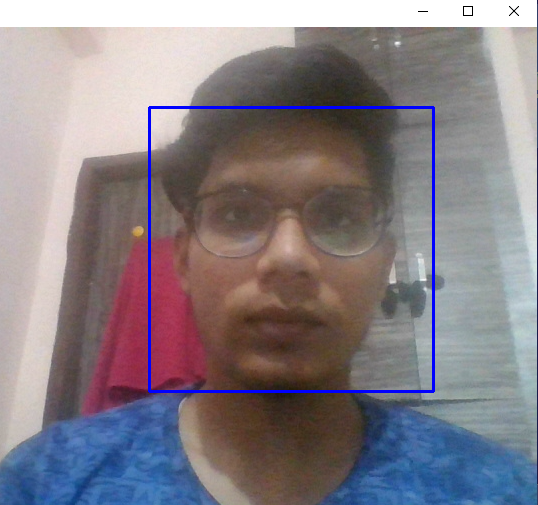
1. **DATA COLLECTION**: 

**Fig.5. Face Recognizer Attendance System GUI**

The student interacts with the system through the Graphical User Interface (GUI) above. The first step the student has to enter his details(Name and ID) this details will be stored in a csv file **'StudentDetailss.csv'**, the ID Number on the GUI. Second step, the student will click on the **TAKE IMAGE** button to capture his faces, here 50pictures of the student will be taken and stored in the **Training Images** Folder. The **haar-cascadeclassifier** file to detect faces through the video stream while the student face is being captured. The notification board will print out the student details after a successful data collection.



**Fig.6. Face Recognizer Attendance System GUI**

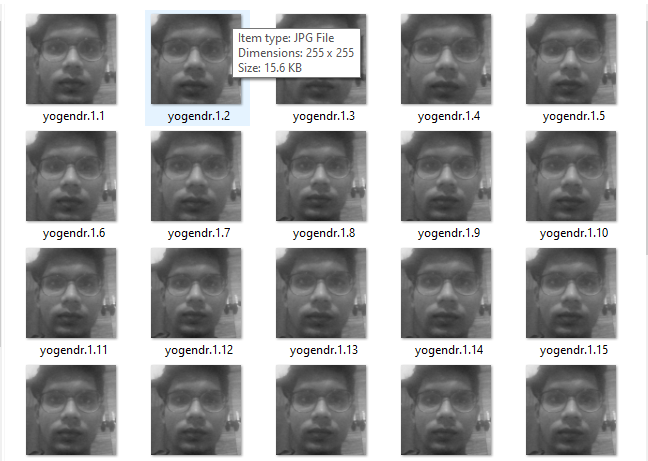


**Fig.7 Face Recognizing**



**Fig.8 System Responds**

Here is our Dataset Sample:



**Fig 9: Dataset sample**

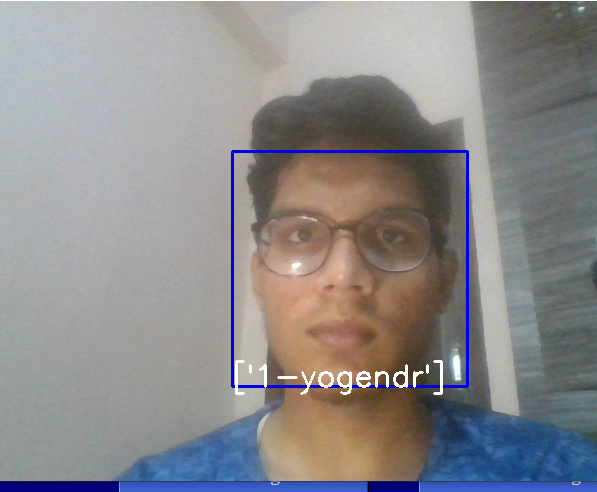
**IMAGE TRAINED**

The student has to click on the **TRAIN IMAGE** button which will link his details, face features to the **LBPHrecognizer** to ease further face recognition, the recognizer will save the face features in the **trainner.yml** and "IMAGE TRAINED" will be printed on the GUI notification board after a successful linkage.



**2. FACE TRACKING**

The student has to click on the **TRACK IMAGE** button to allow the face recognizer to track his face through a video stream, when the system successfully recognize the student face, his details will show and "ATTENDANCE UPDATED" will be printed out otherwise , the ID will be Unknown and "ID UNKOWN, ATTENDANCE NOT UPDATED" will be printed out. Simultaneously, a csv file **AttendanceFile.csv'** will be updated with the ID,NAME of the student and DATE And TIME at which his face has recognized. the Unknown face captured will be store in the **Unknown Images** folder.



**Fig 10: System Recognizing the face**



**Fig 11: Output**

**Pseudo Code:**

1. SET window TO tk.Tk()
2. Window title= "Face\_Recogniser"
3. Window size = ('1580x720')
4. Window configure = (background='dark blue')
5. window.grid\_rowconfigure(0, weight=1)
6. window.grid\_columnconfigure(0, weight=1)
7. SET message TO tk.Label(window, text="Smart Attendance System based on Facial Recognition", bg="royal blue",fg="mint cream", width=50, height=3, font=('times', 30, 'italic bold underline'))
8. message.place(x=200, y=20)
9. SET label TO tk.Label(window, text="Enter ID", width=20, height=2,fg="mint cream", bg="royal blue", font=('times', 15, ' bold '))
10. window size = (x=400, y=200)
11. SET text TO tk.Entry(window, width=20, bg="royal blue",fg="mint cream", font=('times', 15, ' bold '))
12. text.place(x=700, y=215)
13. SET label2 TO tk.Label(window, text="Enter Name", width=20, fg="mint cream", bg="royal blue", height=2, font=('times', 15, ' bold '))
14. label2.place(x=400, y=300)
15. SET txt2 TO tk.Entry(window, width=20, bg="royal blue",fg="mint cream", font=('times', 15, ' bold '))
16. text2.place(x=700, y=315)
17. SET label3 TO tk.Label(window, text="Notification : ", width=20, fg="mint cream",bg="royal blue", height=2, font=('times', 15, ' bold underline '))
18. label3.place(x=400, y=400)
19. SET message TO tk.Label(window, text="", bg="royal blue", fg="mint cream", width=50, height=2, activebackground="royal blue", font=('times', 15, ' bold '))
20. message.place(x=700, y=400)
21. SET-label3-TO-tk,Label(window,text=”Attendence:”,width=20,fg=”mint cream”,bg=”royalblue", height=2, font=('times', 15, ' bold underline'))
22. label3.place(x=400, y=650)
23. SETmessage2TOtk.Label(window,text="",fg="mintcream",bg="royalblue",activeforeground="green", width=60, height=2, font=('times', 15, ' bold '))
24. message2.place(x=700, y=650)
25. DEFINE FUNCTION clear():
    1. txt.delete(0, 'end')
    2. SET res TO ""
    3. message.configure(text=res)
26. DEFINE FUNCTION clear2():
    1. txt2.delete(0, 'end')
    2. SET res TO ""
    3. message.configure(text=res)
27. DEFINE FUNCTION is\_number(s):
    1. TRY:
       1. float(s)
       2. RETURN True
    2. except ValueError:
       1. PASS
28. TRY:
    1. IMPORT unicodedata
    2. unicodedata.numeric(s)
    3. RETURN True
29. except (TypeError, ValueError):
    1. PASS
    2. RETURN False
30. DEFINE FUNCTION TakeImages():
    1. SET Id TO (txt.get())
    2. SET name TO (txt2.get())
    3. if(is\_number(Id) and name.isalpha()):
       1. SET cam TO cv2.VideoCapture(0)
       2. SET harcascadePath TO "haarcascade\_frontalface\_default.xml"
       3. SET detector TO cv2.CascadeClassifier(harcascadePath)
       4. SET sampleNum TO 0
31. while(True):
    1. SET ret, img TO cam.read()
    2. SET gray TO cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)
    3. SET faces TO detector.detectMultiScale(gray, 1.3, 5)
    4. FOR (x, y, w, h) IN faces:
       1. cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)
32. # incrementing sample number
    * 1. SET sampleNum TO sampleNum+1
33. # saving the captured face IN the dataset folder TrainingImage
    * 1. cv2.imwrite(r"TrainingImage/ "+name + "."+Id + '.' +str(sampleNum) + ".jpg", gray[y:y+h, x:x+w])
34. # display the frame
    1. cv2.imshow('frame', img)
35. # wait FOR 50 miliseconds
    1. IF cv2.waitKey(50) & 0xFF EQUALS ord('q'):
       1. break
36. # break IF the sample number is morethan 30
    1. ELSEIF sampleNum > 30:
       1. break
    2. cam.release()
    3. cv2.destroyAllWindows()
    4. SET res TO "Images Saved FOR ID : " + Id + " Name : " + name
    5. SET row TO [Id, name]
37. # with open(r'StudentDetails/StudentDetails.csv', 'r') as csvFile:
38. # IF not csvFile.read().strip():
39. # with open(r'StudentDetails/StudentDetails.csv', 'a+') as csvFile:
40. SET # writer TO csv.writer(csvFile)
41. # writer.writerow(['Id', 'Name'])
    1. with open(r'StudentDetails/StudentDetails.csv', 'a+') as csvFile:
    2. with open(r'StudentDetails/StudentDetails.csv', 'r') as csvFile1:
    3. IF not csvFile1.read().strip():
    4. # with open(r'StudentDetails/StudentDetails.csv', 'a+') as csvFile:
       1. SET writer TO csv.writer(csvFile)
       2. writer.writerow(['Id', 'Name'])
       3. SET writer TO csv.writer(csvFile)
       4. writer.writerow(row)
       5. csvFile.close()
       6. message.configure(text=res)
    5. ELSE:
       1. if(is\_number(Id)):
       2. SET res TO "Enter Alphabetical Name"
       3. message.configure(text=res)
    6. if(name.isalpha()):
       1. SET res TO "Enter Numeric Id"
       2. message.configure(text=res)
42. DEFINE FUNCTION TrainImages():
    1. SET #recognizerTO cv2.face.LBPHFaceRecognizer\_create()#$cv2.createLBPHFaceRecognizer()
    2. SET recognizer TO cv2.face\_LBPHFaceRecognizer.create()
    3. SET harcascadePath TO "haarcascade\_frontalface\_default.xml"
    4. SET detector TO cv2.CascadeClassifier(harcascadePath)
43. SET faces, Id TO getImagesAndLabels("TrainingImage")
    1. recognizer.train(faces, np.array(Id))
    2. recognizer.save(r"TrainingImageLabel/Trainner.yml")
44. SET res TO "Image Trained" # +",".join(str(f) FOR f IN Id)
45. message.configure(text=res)
46. DEFINE FUNCTION getImagesAndLabels(path):
47. # get the path of all the files IN the folder
    1. SET imagePaths TO [os.path.join(path, f) FOR f IN os.listdir(path)]
48. # OUTPUT(imagePaths)
49. # create empth face list
    1. SET faces TO []
50. # create empty ID list
    1. SET Ids TO []
51. # now looping through all the image paths and loading the Ids and the images
52. FOR imagePath IN imagePaths:
53. # loading the image and converting it to gray scale
    1. SET pilImage TO Image.open(imagePath).convert('L')
54. # Now we are converting the PIL image into numpy array
    1. SET imageNp TO np.array(pilImage, 'uint8')
55. # getting the Id from the image
    1. SET Id TO int(os.path.split(imagePath)[-1].split(".")[1])
56. # extract the face from the training image sample
    1. faces.append(imageNp)
57. Ids.append(Id)
58. RETURN faces, Ids
59. DEFINE FUNCTION TrackImages():
    1. SET recognizer TO cv2.face.LBPHFaceRecognizer\_create()
    2. # cv2.createLBPHFaceRecognizer()
    3. recognizer.read(r"TrainingImageLabel/Trainner.yml")
    4. SET harcascadePath TO "haarcascade\_frontalface\_default.xml"
    5. SET faceCascade TO cv2.CascadeClassifier(harcascadePath)
    6. SET df TO pd.read\_csv(r"StudentDetails/StudentDetails.csv")
    7. SET cam TO cv2.VideoCapture(0)
    8. SET font TO cv2.FONT\_HERSHEY\_SIMPLEX
60. SET col\_names TO ['Id', 'Name', 'Date', 'Time']
61. SET attendance TO pd.DataFrame(columns=col\_names)
62. WHILE True:
    1. SET ret, im TO cam.read()
    2. SET gray TO cv2.cvtColor(im, cv2.COLOR\_BGR2GRAY)
    3. SET faces TO faceCascade.detectMultiScale(gray, 1.2, 5)
63. for(x, y, w, h) IN faces:
    1. cv2.rectangle(im, (x, y), (x+w, y+h), (225, 0, 0), 2)
    2. SET Id, conf TO recognizer.predict(gray[y:y+h, x:x+w])
    3. if(conf < 50):
       1. SET ts TO time.time()
       2. SET date TO datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
       3. SET timeStamp TO datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
       4. SET aa TO df.loc[df['Id'] EQUALS Id]['Name'].values
       5. SET tt TO str(Id)+"-"+aa
       6. SET attendance.loc[len(attendance)] TO [Id, aa, date, timeStamp]
    4. ELSE:
       1. SET Id TO 'Unknown'
       2. SET tt TO str(Id)
       3. if(conf > 75):
          1. SET noOfFile TO len(os.listdir("ImagesUnknown"))+1
          2. cv2.imwrite(r"ImagesUnknown/Image" +
          3. str(noOfFile) + ".jpg", im[y:y+h, x:x+w])
          4. cv2.putText(im, str(tt), (x, y+h), font, 1, (255, 255, 255), 2)
          5. SET attendance TO attendance.drop\_duplicates(subset=['Id'], keep='first')
          6. cv2.imshow('im', im)
64. IF (cv2.waitKey(1) EQUALS ord('q')):
    1. break
65. SET ts TO time.time()
66. SET date TO datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
67. SET timeStamp TO datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
68. SET Hour, Minute, Second TO timeStamp.split(":")
69. SET-fileName-TO-r"Attendance/Attendance\_"+date+"\_"+Hour+"-"+Minute+"-"+Second+".csv"
70. attendance.to\_csv(fileName, index=False)
71. cam.release()
72. cv2.destroyAllWindows()
73. OUTPUT(attendance)
74. SET res TO ''
    1. FOR i IN attendance:
75. #OUTPUT(i)
76. res += (''.join(str(attendance[i])))
77. #OUTPUT(res)
78. #OUTPUT(''.join(res))
79. message2.configure(text=res)
80. DEFINE FUNCTION quit():
    1. SET dialog\_title TO 'QUIT'
    2. SET dialog\_text TO 'Are you sure?'
    3. SET answer TO messagebox.askquestion(dialog\_title, dialog\_text)
    4. IF answer EQUALS 'yes':
       1. window.destroy()
81. SET clearButton TO tk.Button(window, text="Clear", command=clear, fg="mint cream", bg="royal blue",width=20, height=2, activebackground="mint cream", font=('times', 15, ' bold '))
82. clearButton.place(x=1050, y=200)
83. SET clearButton2 TO tk.Button(window, text="Clear", command=clear2, fg="mint cream", bg="royal blue", width=20, height=2, activebackground="mint cream", font=('times', 15, ' bold '))
84. clearButton2.place(x=1050, y=300)
85. SET takeImg TO tk.Button(window, text="Take Images", command=TakeImages, fg="mint cream", bg="royal blue", width=20, height=3, activebackground="mint cream", font=('times', 15, ' bold '))
86. takeImg.place(x=200, y=500)
87. SET trainImg TO tk.Button(window, text="Train Images", command=TrainImages, fg="mint cream",bg="royal blue", width=20, height=3, activebackground="mint cream", font=('times', 15, ' bold '))
88. trainImg.place(x=500, y=500)
89. SET trackImg TO tk.Button(window, text="Track Images", command=TrackImages, fg="mint cream",bg="royal blue", width=20, height=3, activebackground="mint cream", font=('times', 15, ' bold '))
90. trackImg.place(x=800, y=500)
91. SET quitWindow TO tk.Button(window, text="Quit", command=quit, fg="mint cream", bg="royal blue", width=20, height=3, activebackground="mint cream", font=('times', 15, ' bold '))
92. quitWindow.place(x=1100, y=500)
93. window.mainloop()

## **3.5 Steps to run the project**

* Open the project in any editor like PyCharm , Visual Studio etc
* When we run **train.py** a window is opened and asks for Enter Id and Enter Name. After enter name and id then we have to click Take Images button.
* By clicking Take Images camera of running computer is opened and it start taking image sample of person.
* This Id and Name is stored in folder Student Details and file name is StudentDetails.csv.
* It takes 60 images as sample and stores them in folder Training Image. After completion it notifies that images saved.
* After taking image sample we have to click Train Image button. Now it take few seconds to train machine for the images that are taken by clicking Take Image button and creates a Trainner.yml file and store in Training Image Label folder.
* Now all initial setups are done. By clicking Track Image button camera of running machine is opened again. If face is recognized by system then Id and Name of person is shown on Image. Press Q (or q) for quit this window. After quitting it attendance of person will be stored in Attendance folder as csv file with name, id, date and time and it is also available in window.

## **3.6 Result analysis**

* The interface for the Smart Attendance System has been created. Using the interface the images of the individual students is being recorded and stored in the training dataset. Simultaneously their information is stored in the database i.e. excel sheet. Finally the images of the students is being tracked and recognized.
* Fig 4. The different folders have been created.
* Fig 5. The interface for the Face Recognition Based Attendance System in which the Id and Name of the respective students are stored.
* Fig 6. The images are stored in a folder named Training Images.
* Fig 7. The excel sheet for the student details is created.
* Fig 8. The names of the students have been stored in the Student Details excel sheet.
* Fig 9. The images of the students are trained.
* Fig 10. After tracking the images are attendance of the students is marked.
* Fig 11. The excel sheet for attendance of the students is created.
* Fig 12. The students attendance record is stored in the excel sheet

# Conclusion

Smart attendance management system is designed to solve the issues of existing manual systems. We have used face recognition concept to mark the attendance of student and make the system better. The system performs satisfactory in different poses and variations. In future this system need be improved because these systems sometimes fails to recognize students from some distance, also we have some processing limitation, working with a system of high processing may result even better performance of this system.

This paper features the most productive Open CV face recognition method accessible for Attendance Management. The system has been implemented using the LBPH algorithm. LBPH excels other algorithms by confidence factor of 2-5 and has least noise interference. The implementation of the Smart Attendance System portrays the existence of an agreement between the appropriate recognition rate and the threshold value. Therefore LBPH is the most authentic and competent face recognition algorithm found in Open CV for the identification of the students in an educational institute and marking their attendance adequately by averting proxies.

# Future Scope

Project targets the students of different academic levels and faculty members. The main constraint we faced is distinguishing between identical twins. This situation is still a challenge to biometric systems especially facial recognition technology. According to Phillips and his co-researcher paper to get the best results of the algorithms your system employed, they should run under certain conditions for taken pictures (i.e., age, gender, expressions, studio environmental etc.) otherwise, the problem is still ongoing.

They provide application (method) to solve this problem, but in order to use this solution you have to sign a contract with the (NIST) organization and to be a researcher or developer. For us, to solve this issue we suggest to record twins’ attendance manually.

# 6. References

1)Smart Attendance System using Computer Vision and Machine Learning Dipti Kumbhar#1 , Prof. Dr. Y. S. Angal\*2 # Department of Electronics and Telecommunication, BSIOTR, Wagholi, Pune,

2) Face Recognition based Attendance Management System using Machine Learning Anushka Waingankar1, Akash Upadhyay2, Ruchi Shap, Nevil Pooniwala4, Prashant Kasambe5

3) Face Recognition Based Student Attendance System with OpenCV CH. VINOD KUMAR1 , DR. K. RAJA KUMAR2 1 PG Scholar, Dept of CS& SE, Andhra University, Vishakhapatnam, AP, India. 2Assistant Professor, Dept of CS& SE, Andhra University, Vishakhapatnam, AP, India.

1. <https://www.superdatascience.com/blogs/opencv-face-recognition>
2. [https://towardsdatascience.com/face-recognition-how-lbph-works- 90ec258c3d6](https://towardsdatascience.com/face-recognition-how-lbph-works-%2090ec258c3d6)b
3. [https://www.pyimagesearch.com/2018/09/24/opencv-face- recognition/](https://www.pyimagesearch.com/2018/09/24/opencv-face-%20recognition/)
4. <http://www.ijsrp.org/research-paper-0218/ijsrp-p7433.pdf>
5. https://iopscience.iop.org/article/10.1088/1757- 899X/263/4/042095/pdf