Methodologies.

Proposed arrangement for system design

Conclusion

Acknowledgement

Reference

**7. Proposed arrangement for system design**

**7.1 Tools Required**

**SOFTWARE TOOLS**

There are several **software** tools and technologies that can be used for developing a smart attendance management system using machine learning. Here are the ones that we have used:

1. ***Python***: Python is a popular programming language for data science and machine learning. It has a vast number of libraries and tools that can be used for machine learning tasks.
2. ***OpenCV***: OpenCV is an open-source computer vision library that can be used for tasks such as face recognition, object detection, and image processing.
3. ***MySQL***: MySQL is a popular relational database management system that can be used for storing attendance data and managing the database.
4. ***VS Code***: VS Code are popular integrated development environments (IDEs) for Python. They can be used for writing and debugging code.
5. ***NumPy***: NumPy (Numerical Python) is a popular open-source Python library that is widely used for scientific computing and data analysis.

**HARDWARE TOOLS**

The **hardware** tools that will be needed to complete this project are:

1. ***Cameras:*** High-resolution cameras can be used for capturing attendance data and for tasks such as face recognition.
2. ***Display Units:*** Display units such as LED panels or LCD screens can be used for displaying attendance data, notifications, and alerts.
3. ***Power Supply:*** Power supply units such as batteries or power adapters can be used for powering the devices and hardware components.

**7.2 Overview Of architecture**

**7.3 User Interface**

The smart attendance management system has been designed to be easy to use and navigate, with a simple and intuitive interface that can be used by people without prior technical knowledge. The main screen of the system features eight clearly labelled buttons that represent different functions of the software, including managing student details, training the system with facial data, taking attendance, displaying attendance records, accessing developer information, contacting the help desk, and exiting the application.

The buttons are labelled with icons to provide a visual cue for users. The icons are designed to be easily recognizable and to represent the different functions of the software in a clear and concise manner.

**7.4 Collection of Student Details:**

Student details like name, address, email-ID, phone number is taken and a unique ID is given to each student.

100 images corresponding to each student is taken for training the model.

**7.5 Pre-processing:**

* ***CONVERSION OF IMAGES TO GRAYSCALE***

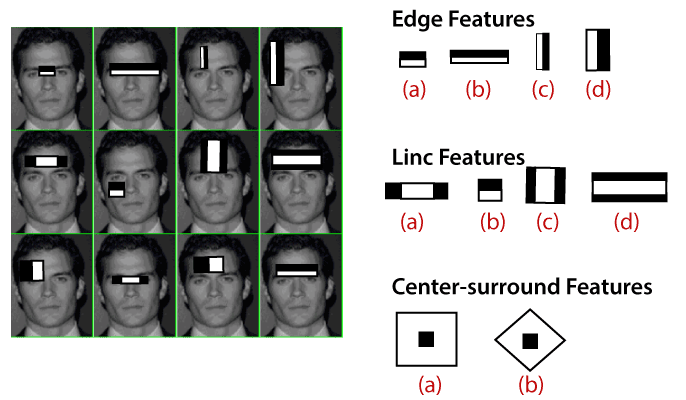
1. **Simplicity:** Working with grayscale images is simpler than working with coloured images. A grayscale image only has one channel, whereas a colored image has three (red, green, and blue). This means that we can perform the same face detection and recognition tasks using only a single channel of information, which simplifies the algorithms and makes them more efficient.
2. **Reduced Noise:** The grayscale conversion removes the color information, which can help to reduce the amount of noise in the image. By removing colour, the image data is less complex, and there is less chance of having noisy pixels that could interfere with the detection process.
3. **Increased Contrast:** Converting to grayscale can help to increase the contrast of the image, making it easier to detect features such as edges and lines that are important for face detection and recognition.

***CONVERSION OF GRAY SCALE IMAGES TO NUMPY ARRAY:***

1. Converting an image to a NumPy array is necessary when working with OpenCV because it allows us to take advantage of the powerful NumPy array operations and functions that OpenCV provides.
2. The NumPy array representation also provides a convenient way to store and pass image data between different functions and modules in our code.

**7.6 FEATURE EXTRACTION:**

In this step, we extract the features from the image, with the help of edge detection, line detection, and centre detection. Then provide the coordinate of x, y, w, h, which makes a rectangle box in the picture to show the location of the face. It can make a rectangle box in the desired area where it detects the face.



**7.8 ALGORITHMS USED**

In the HAAR cascade, a cascade function is trained using a large number of both positive and negative pictures. Images with faces are considered positive, whereas those without faces are considered negative. Image characteristics are viewed in face detection as numerical data taken from the images that can differentiate one image from another.

On every training image, we run every algorithm feature. At first, each image is given equal weight. It discovered the most accurate threshold for classifying faces as positive or negative. Errors and incorrect categorization could exist. We choose the features with the lowest error rate, i.e., the features that classify face and non-face images the most accurately.

**7.9 LBPH algorithm**

All things considered and taking into account the region, Local Binary Pattern (LBP) can be a clear yet effective surface that leaves marks on certain areas of the image by obstructing the area.

as a paired width as a result. It was determined that the securing execution of more informational indexes is improved when LBP is combined with histograms of characterised angles (HOG) directed. We can represent to confront images with a vector of direct information using LBP combined with histograms.

The LBPH algorithm is composed of the following 5 steps:

**i. The LBPH employs 4 parameters:**

**Radius**: The radius is a local binary pattern-making tool that depicts the area surrounding the centre pixel. One is chosen as the frequency.

**Neighbours**: the quantity of sample points needed to create a binary's circular region. Remember that the cost of the computer increases as you add more points. It is tuned to 8 cycles per second.

**Grid X**: the quantity of cells arranged horizontally. The better the grid, the bigger the vector size of the emerging element, and the more cells. It's set to 8 cycles per second.

**Grid Y:** refers to how many cells are in a straight line.

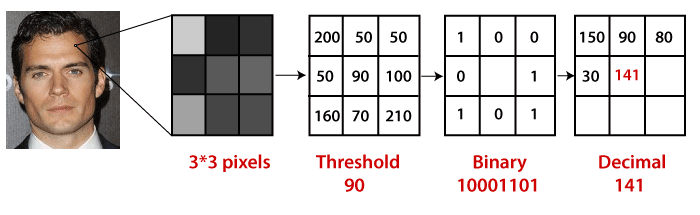
The better the grid, the bigger the vector size of the emerging element, and the more cells. It's set to 8 cycles per second.

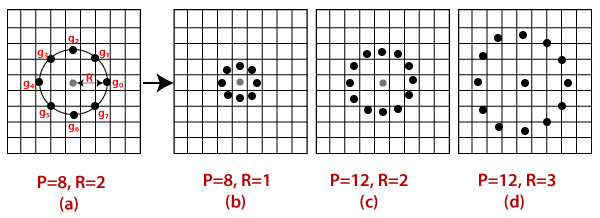
**ii. Algorithm Training**

The model must first be trained. We achieve this by using a database that includes pictures of the people we want to tell. The next step is to add an ID (also known as a number or a name) to each image so that the algorithm can use this information to identify the input image and output the result. The ID for each image of the same individual must match. Let's examine the LBPH process phases since the training set has already been made.

**iii. Applying for LBP operation**

The next step is to develop an intermediary image that clarifies the initial photograph taken on a roadway with face light. Algorithmic law does this by utilising a window view that supports a variety of frames and neighbours.

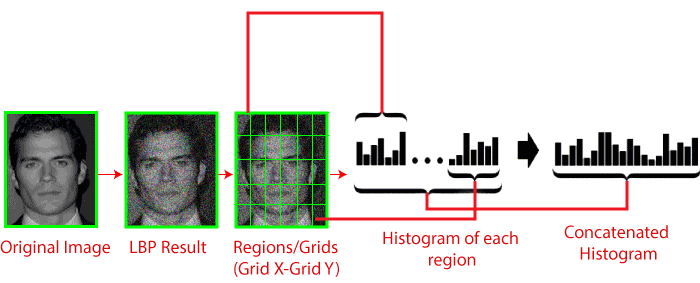


* We have a photo of a grey face.
* This image can be represented in part as a 3x3 pixel window or as a 3x3 matrix holding the range of pixel thicknesses (0 to 255).
* After that, we must determine the matrix's median value, which will serve as the limit and be utilised to characterise fresh values from eight neighbours.
* We establish a new binary value for each median of the middle value (limit). We set 0 values below the limit and 1 value greater than the limit.
* Now, regardless of average value, the matrix will only include binary values. Line by line, we must estimate each binary value from each place in the matrix to the new binary value (for example, 10001101). Note that while some authors synchronise binary values using different techniques (such as clock direction), the outcome will always be the same.
* This binary number is then converted to a decimal value and set to the matrix's centre value, which is a pixel from the first image.
* After this procedure (LBP process), we obtain a new image that more accurately captures the characteristics of the original image.

Bilinear interpolation is an option for doing this. When determining the location of a new data point, values from the four (2x2) closest pixels are used if the new data point is within one pixel.

**iv. Recording histogram:**

Now, using the image created in the last step, we can use the Grid X and Grid Y parameters to split the image into multiple grids, as can be seen in the following picture:



Each histogram (from each grid) will only have 256 (0 255) positions because our image is grey; these places reflect the potential for each pixel power. Then, in order to construct a new and larger histogram, we must sync each histogram. The final histogram will have 8x8x256 = 16.384 places if we assume that we have 8x8 grids. The characteristics of the initial image are depicted in the final histogram.

**V. Performing face recognition:**

The algorithm has already been trained at this point. Each histogram produced serves as a representation of a different image from the training database. As a result, whenever we are given an image to insert, we repeat the steps and produce a histogram for the new image.

- Thus, all that is required to create an image that is comparable to an input image is to compare the two histograms and replace the image with the histogram that comes closest.

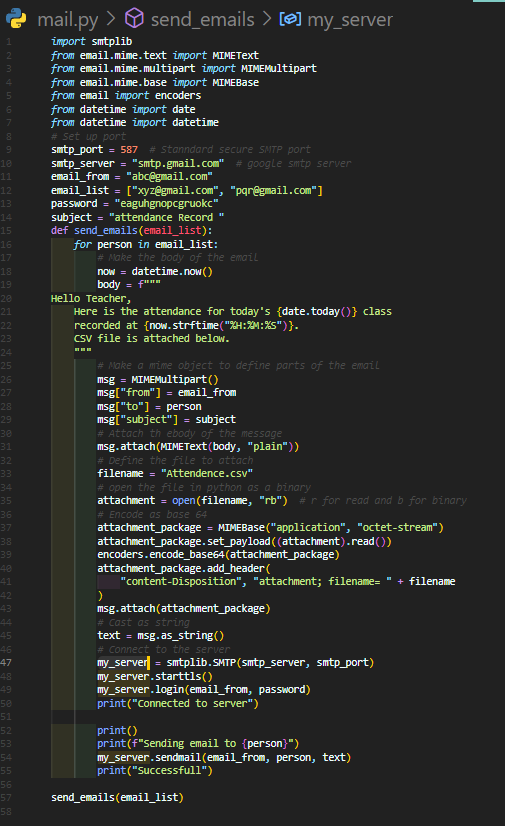
Face recognition and Face detection- Different techniques, such as the Euclidean distance, square-chi, total value, etc., can be used to compare two histograms and determine their distance from one another. Based on the following calculation, we can utilise the Euclidean (most common) range in this example:

**Distance Between two histograms**

* So, the output of the calculation is an ID from a picture with a close-by histogram. The calculation ought to likewise restore a determined reach, which can be utilized as a proportion of 'certainty'. Note: don't be tricked by the word 'certainty', since low camouflage is better since it implies that the separation between the two histograms is nearer.
* We can then use the threshold limit and 'confidence' to automatically adjust if the algorithm has detected the image correctly. We can assume that the algorithm has detected success when confidence is below the defined limit.

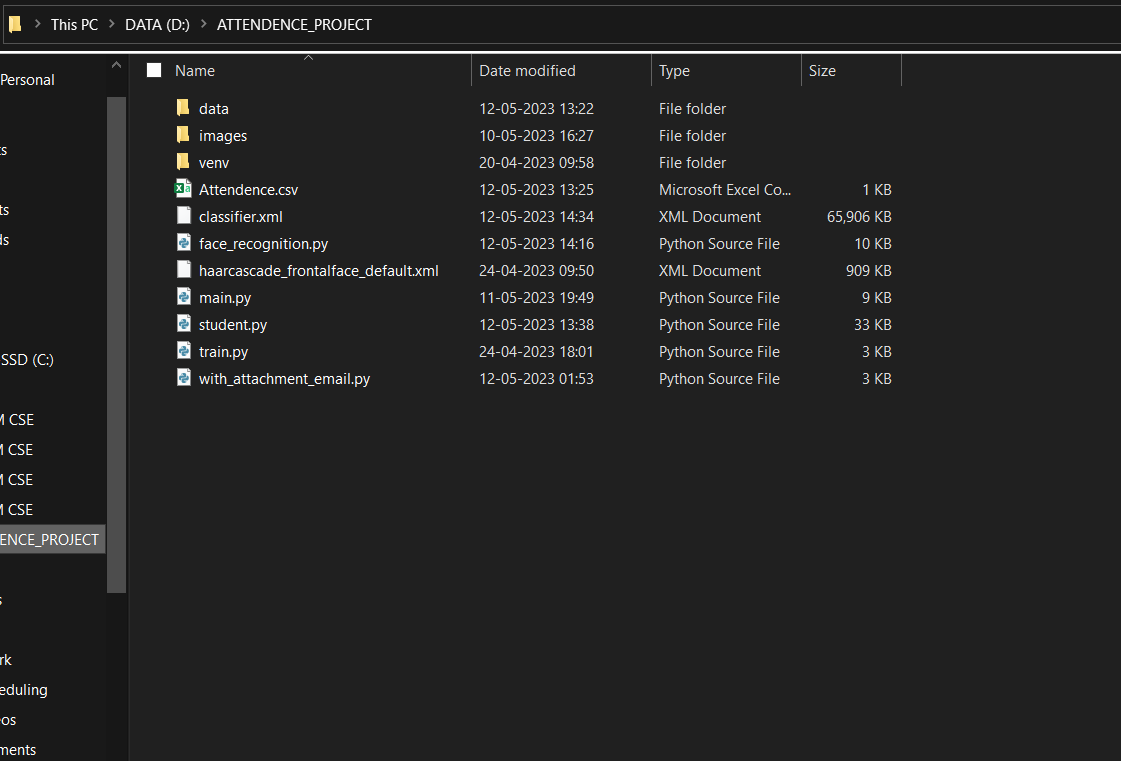
**7.10 Mail Notification to teacher**

In this project, we have added an extra feature called auto-mail. It automatically sends the attendance file to specific mail (specified in the system model). The code for the same is given below:



**8. Overview:**

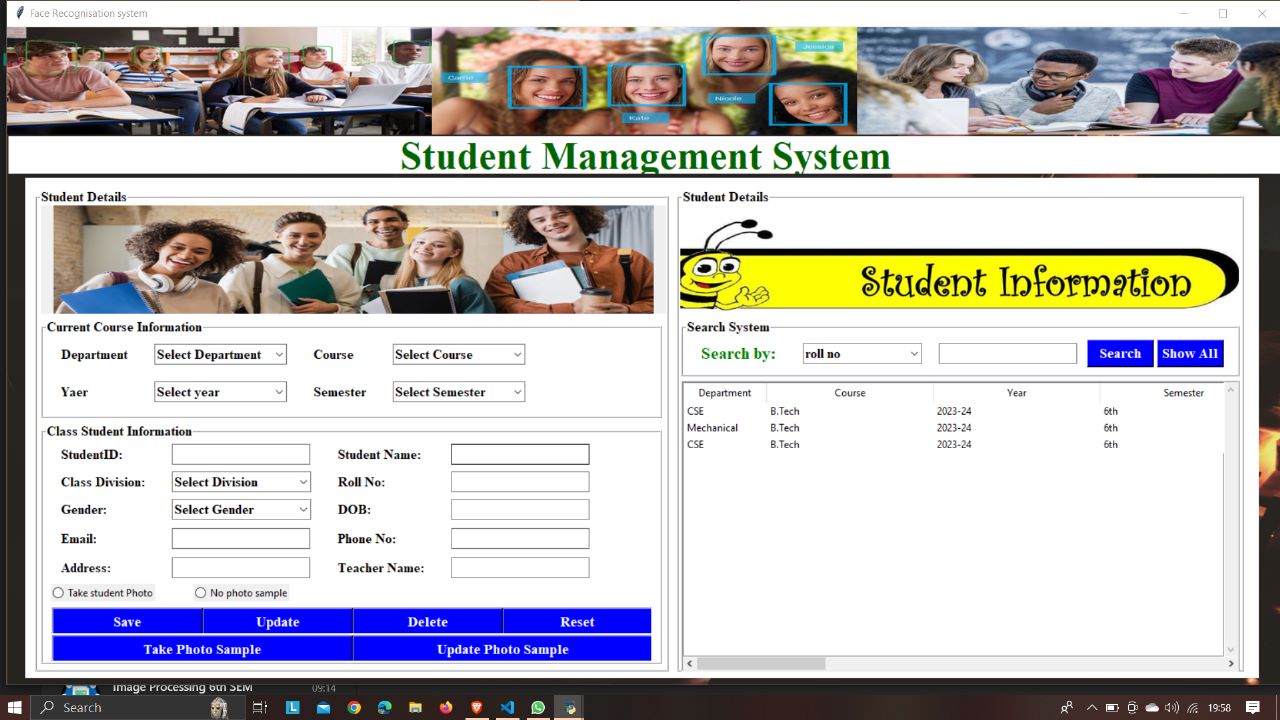
1. "Students Details" button: This button will allow you to view and manage the details of all registered students, such as their names, IDs, and photos, which are stored in the database.
2. "Train Data" button: This button is used to train the system with facial data of the registered students so that the system can recognize them accurately during attendance taking.
3. "Take Attendance" button: This button initiates the face recognition process to capture students' images during a class session and then compares the images with the previously stored data to mark attendance automatically.
4. "Display Attendance" button: This button will display the attendance data of students who were present, absent, or late for a particular class session or for an entire semester.
5. "Developers" button: This button will provide information about the developers who built the system, including their names, photos, and contact details.
6. "Help Desk" button: This button will allow users to access the system's support team for any assistance or technical issues they may encounter while using the system.
7. “Photos” button: It shows all the collected photos of students.
8. "Exit" button: This button will exit the attendance management system and close the application.

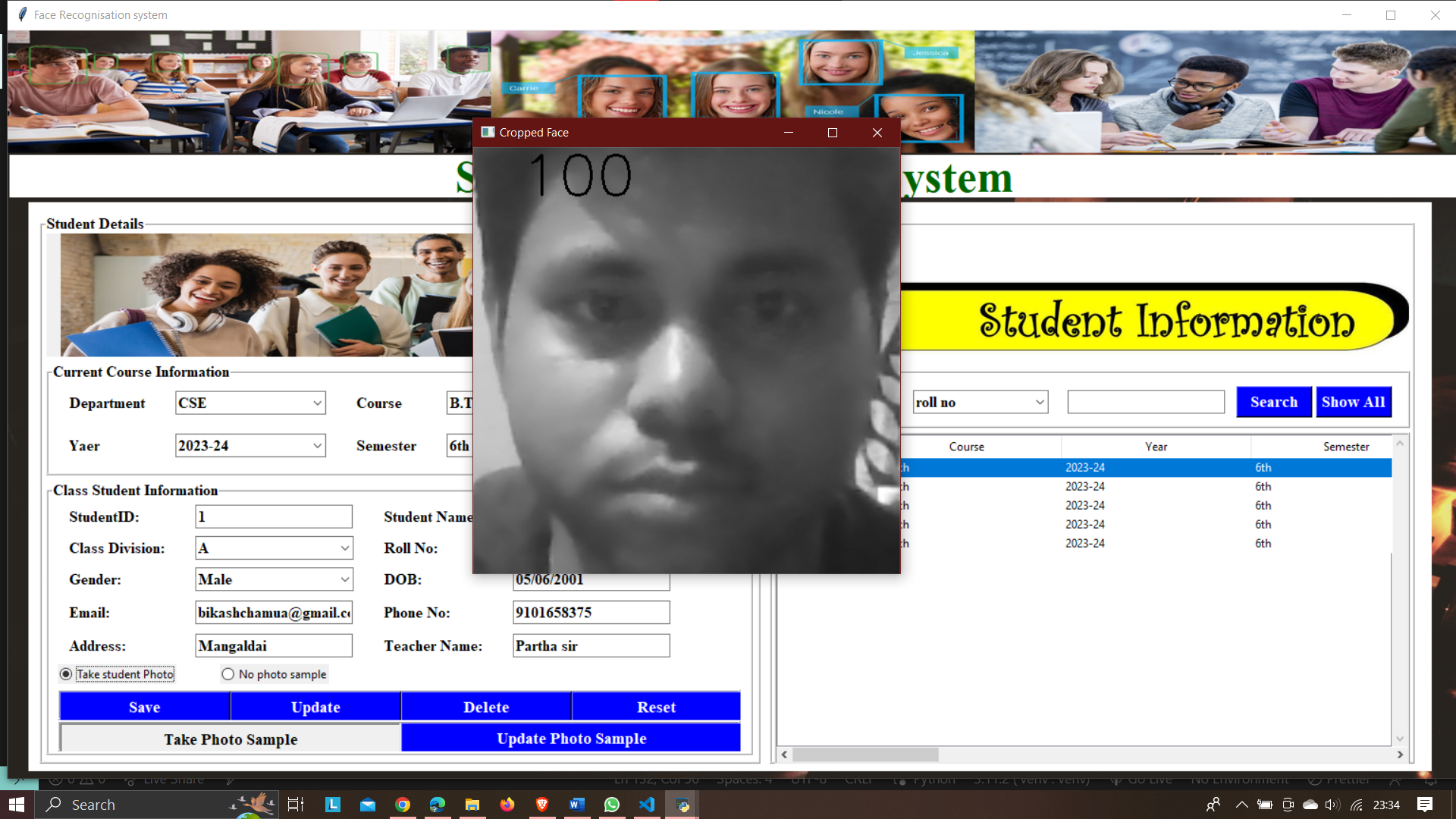


**Fig 8.1: File Architecture**



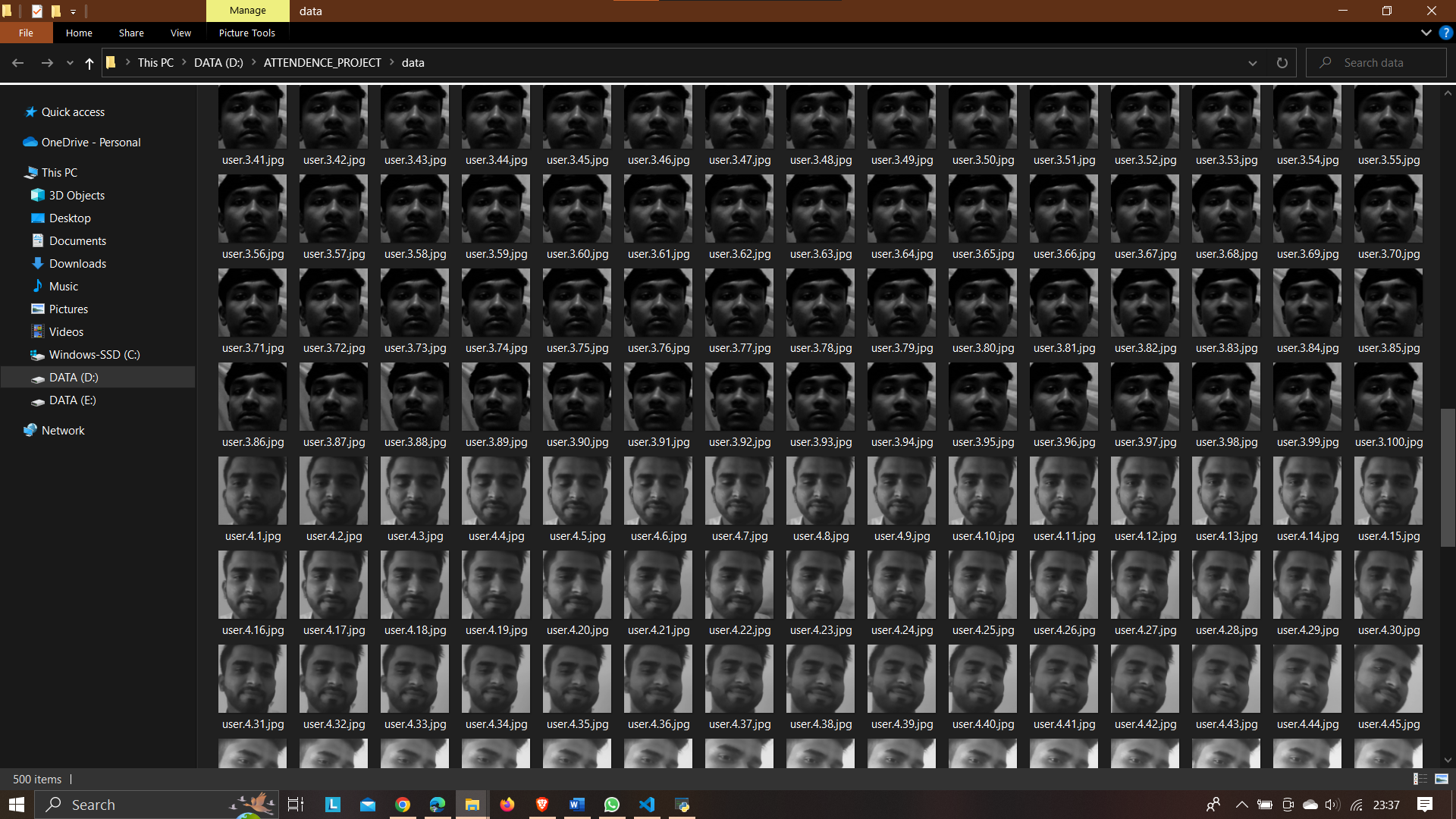
**Fig 8.2: Home Page**

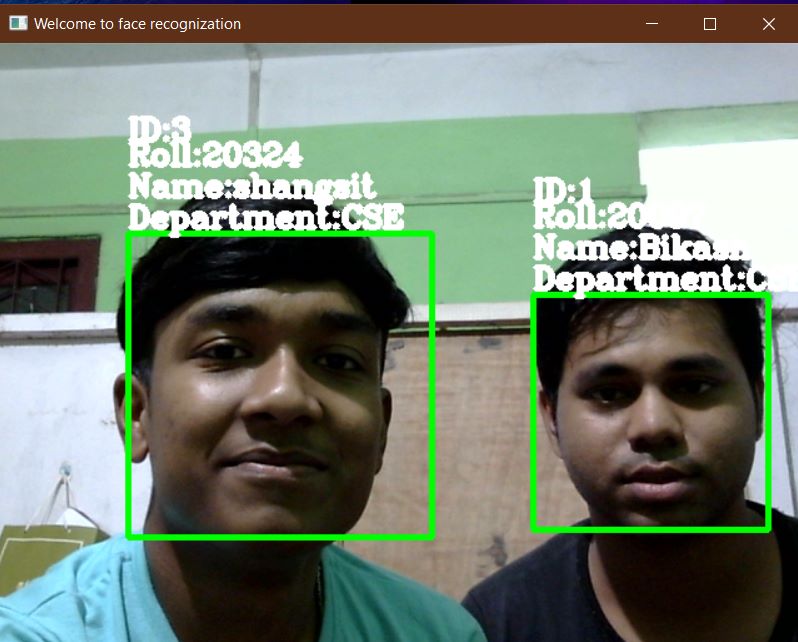
**fig 8.3: Students’ Information**

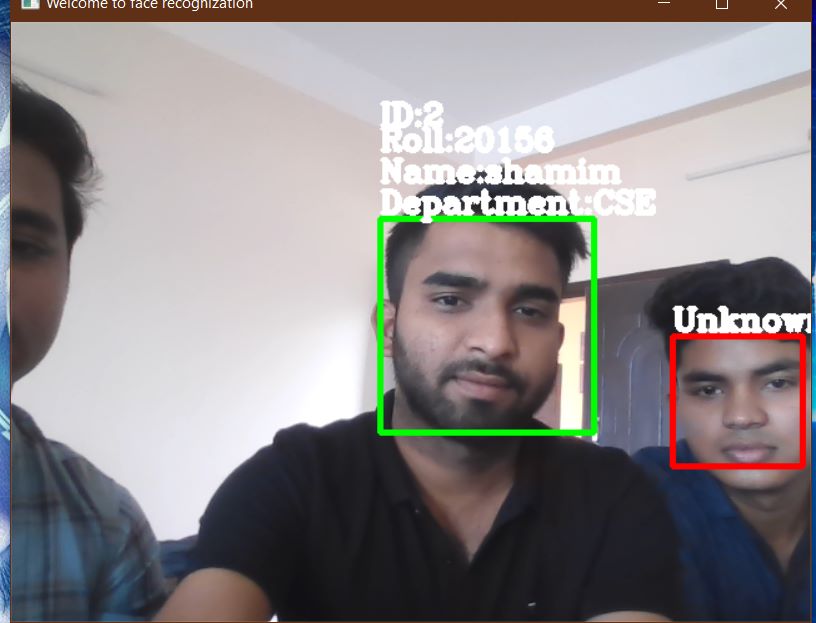
****

**Fig: 8.4 Take Student’s image**

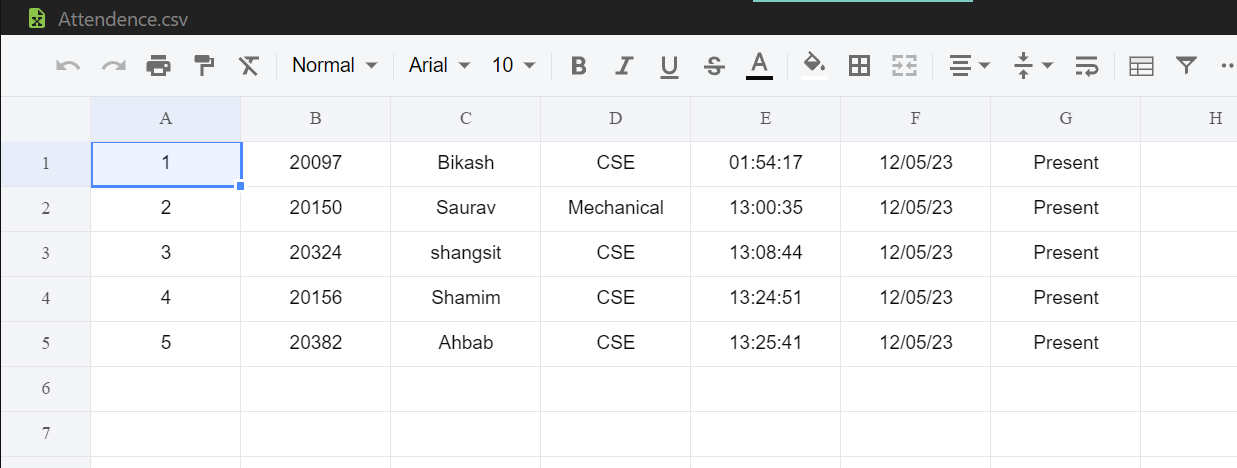
**8.5: Training Images**

**fig 8.6 Photo Samples with unique id**

**Fig 8.7 Recognizing trained faces**



**Fig 8.8 Marking unknown faces**

****

**8.9 Attendance in the csv file with time stamp**

**9. Conclusion and Future scope:**

The entire project has been produced, together with evaluation and testing, from the specifications to a complete system. The system that was created has succeeded in its goal and aims. A project intrinsically requires more thorough examination. The techniques can be coupled with others to produce pleasing outcomes. According to the literature review, completely different approaches have been used in the past. Due to the problems found in the system's recognition component, it was decided to base the system's parameters on a relatively limited class size.

For security purposes, the system would be need to impose login practicality. Data privacy is really essential. The organisation can capture and store pictures of the newest members at the start of each year.

Everyone will have the right to know whether their faces will be used in a face recognition attendance system. Government policies on moral issues, as well as laws and rights relating to information protection, should be in accordance with this.

The people can give their permission for their photos to be used for attendance purposes.

The interaction between our technology, the students, and the teachers will aid us in future work to increase the accuracy of face detection. The desegregation video-streaming service and lecture archiving system, on the other hand, enhance our system and provide more significant applications in the areas of remote education, course management systems (CMS), and support for college development (FD). By running this system with more students sitting on a bench and allowing them to switch places, we will be able to perfect this approach.

**10. Acknowledgement:**

A big thank you to Prof. Partha Bhowmik for his support and guidance throughout this project. We gained so much knowledge about software engineering processes that we are very thankful to our professor. We also thank our friends for their help and encouraging words. Most of all a big thanks to God Almighty for his guidance throughout the course. Thank you all for your encouragement and support!