PYTHON PROJECT

TITLE: FACE RECOGNITION BASED ATTENDANCE SYSTEM

1. INTRODUCTION

Face recognition is a broad challenge of verifying or identifying people in pictures or videos. Big tech giants are still working to make a faster and more accurate face recognition model.

A face recognition based attendance system uses a camera to capture an image of a person's face and then uses software to compare it to a database of known faces in order to verify the person's identity. This technology can be used to automate the process of taking attendance in a classroom or office setting.

The system typically includes a camera for capturing images, a computer for running the recognition software, and a database for storing information about known faces. The system may also include a display for showing the status of the attendance and a means for recording the attendance data, such as a spreadsheet or database.

To implement a face recognition based attendance system, the following steps can be taken:

1. Gather a dataset of images of faces of the individuals who will be using the system.
2. Train a face recognition model using the dataset.
3. Integrate the trained model into the attendance system.
4. Test the system to ensure it is accurately identifying individuals.
5. Deploy the system in the desired location.

Advantages of using a face recognition based attendance system include increased accuracy and efficiency compared to traditional methods, such as manual sign-in sheets. Additionally, this technology can help to reduce the spread of infectious diseases by eliminating the need for physical contact with shared surfaces.

However, there are also concerns about privacy and the potential for misuse of facial recognition technology. It is important to ensure that the system is implemented in accordance with relevant laws and regulations, and to provide transparent information about how the data collected by the system will be used and protected.

Overall, a face recognition based attendance system can provide a convenient and effective solution for automating attendance tracking, but it is important to consider the potential risks and ensure that the system is implemented responsibly.

Real-World Applications of Face Recognition:
Face recognition is currently being used to make the world safer, smarter, and more convenient.

There are a few use cases :

» Finding Missing Person

» Retail Crime

» Security Identification

» Identifying accounts on social media

» School Attendance System

» Recognizing Drivers in Cars

There are several methods to perform facial recognition depending on the performance and complexity. *Drawbacks of current attendance system:*

Risk of human error
We're all human and we can easily make mistakes when collecting data manually.Employees don't always punch in the correct date and time. You don't always enter the correct numbers into the attendance spreadsheets.You may face situations in which employees come to you asking why they receive less than the hours they've worked, or you may lose thousands of dollars without even knowing it.

Employee time theft
Time theft happens when employees try to get paid for the time they don't work. It can be in the form of buddy punching, clocking in earlier, clocking out later to receive overtime money, etc. You know the consequences: time card frauds lower workplace morale and cost your business a lot of money.

Time-consuming
Collecting time cards, verifying them, and processing them for payroll is surely time-consuming, especially when you manage a larger team.Every time there are errors or issues regarding their attendance, staff will flood in to ask you to make adjustments. Admin work will soon suck at least 5 more hours of your week, which means you've lost over 260 hours every year.If you're leading a team, you certainly have a lot on your plate. Why wasting time on manual attendance tracking when there are other time-saving and convenient solutions out there?

More paperwork
Employee work hours need to be recorded and stored for compliance purpose. If you're still using paper timesheets, paperwork will soon drown you in boredom and irritation.Paper timesheets aren't cheap, take up more space, and are hard to retrieve when needed. This can be a big hassle for managers of businesses with multiple locations and employees.

Reasons to upgrade to automated attendance:

More accurate payroll
Automated attendance software eliminates the risk of human error, hence providing more accurate data for payroll. Employers don't have to be afraid of compliance issues, and employees don't have to be afraid of being paid incorrectly.

Prevent time theft
Attendance tracking software often asks staff to clock in by using fingerprints and location stamps, accessing the location's Wi-Fi, or taking photos of themselves at the job sites. It limits and prevents time theft behaviors such as employees clocking in/out for their friends or punching in/out the wrong time.

Improved engagement, productivity, and efficiency

When staff know your attendance system is accurate and reliable, they don't have to be afraid of being paid inaccurately anymore. They can focus more on work, productivity, and efficiency instead of trying to trick the attendance system.Attendance apps aren't just beneficial for employees' performance, but for managers as well. You can spend less time checking attendance and more on things that require your attention.

More integrated features

There might be other features besides attendance tracking such as scheduling or team messaging. Plus, you can integrate attendance software with HR and payroll software, or have the software tailored to your business needs.

2.RELATED STUDIES

As the part of our project, we learned the following modules of python from the sources specified below.

Tkinter:

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

Source: TkInter - Python Wiki

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.

This OpenCV tutorial will help you learn the Image-processing from Basics to Advance, like operations on Images, Videos using a huge set of Opencv-programs and projects.

Source: OpenCV: Cascade Classifier

Numpy:

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

Array in Numpy is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In Numpy, number of dimensions of the array is called rank of the array.A tuple of integers giving the size of the array along each dimension is known as shape of the array. An array class in Numpy is called as ndarray. Elements in Numpy arrays are accessed by using square brackets and can be initialized by using nested Python Lists.

Source: NumPy: the absolute basics for beginners — NumPy v1.24 Manual

Pandas:

Pandas is an open-source library that is built on top of NumPy library. It is a Python package that offers various data structures and operations for manipulating numerical data and time series. It is mainly popular for importing and analyzing data much easier. Pandas is fast and it has high-performance & productivity for users.

Source: Installation — pandas 1.5.3 documentation (pydata.org)

Csv:

CSV (Comma Separated Values) is a simple file format used to store tabular data, such as a spreadsheet or database. A CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format.

For working CSV files in Python, there is an inbuilt module called csv.

Source: csv — CSV File Reading and Writing — Python 3.11.1 documentation

Openpyxl:

Openpyxl is a Python library for reading and writing Excel (with extension xlsx/xlsm/xltx/xltm) files. The openpyxl module allows Python program to read and modify Excel files.

For example, users might have to go through thousands of rows and pick out a few handful of information to make small changes based on some criteria. Using Openpyxl module, these tasks can be done very efficiently and easily.

source: openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files — openpyxl 3.0.10 documentation

Pillow:

Digital Image processing means processing the image digitally with the help of a computer. Using image processing we can perform operations like enhancing the image, blurring the image, extracting text from images, and many more operations. There are various ways to process images digitally. Here we will discuss the Pillow module of Python. Python Pillow is built on the top of PIL (Python Image Library) and is considered as the fork for the same as PIL has been discontinued from 2011. Pillow supports many image file formats including BMP, PNG, JPEG, and TIFF. The library encourages adding support for newer formats in the library by creating new file decoders.

Source: Handbook - Pillow (PIL Fork) 9.4.0 documentation

3. PROPOSED MODEL/APPLICATION

TECHNOLOGY USED:

- tkinter for whole GUI
- OpenCV for taking images and face recognition (cv2.face.LBPHFaceRecognizer_create())
- CSV, Numpy, Pandas, datetime etc. for other purposes.

FEATURES:

- Easy to use with interactive GUI support.
- Password protection for new person registration.
- Creates/Updates CSV file for deatils of students on registration.
- Creates a new CSV file every day for attendance and marks attendance with proper date and time.
- Displays live attendance updates for the day on the main screen in tabular format with Id, name, date and time.

Face Recognition: Understanding LBPH Algorithm

- **Face Detection**: it has the objective of finding the faces (location and size) in an image and probably extract them to be used by the face recognition algorithm.
- **Face Recognition**: with the facial images already extracted, cropped, resized and usually converted to grayscale, the face recognition algorithm is responsible for finding characteristics which best describe the image.

The face recognition systems can operate basically in two modes:

- **Verification or authentication of a facial image**: it basically compares the input facial image with the facial image related to the user which is requiring the authentication. It is basically a 1x1 comparison.
- **Identification or facial recognition**: it basically compares the input facial image with all facial images from a dataset with the aim to find the user that matches that face. It is basically a 1xN comparison.

There are different types of face recognition algorithms, for example:

- **Eigenfaces (1991)**

- **Local Binary Patterns Histograms (LBPH) (1996)**
- **Fisherfaces (1997)**
- **Scale Invariant Feature Transform (SIFT) (1999)**
- **Speed Up Robust Features (SURF) (2006)**

Each method has a different approach to extract the image information and perform the matching with the input image. However, the methods Eigenfaces and Fisherfaces have a similar approach as well as the SIFT and SURF methods.

Today we gonna talk about one of the oldest (not the oldest one) and more popular face recognition algorithms: **Local Binary Patterns Histograms (LBPH)**.

Objective

The objective of this post is to explain the **LBPH** as simple as possible, showing the method step-by-step.

As it is one of the easier face recognition algorithms I think everyone can understand it without major difficulties.

Introduction

Local Binary Pattern *(LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.*

It was first described in 1994 (LBP) and has since been found to be a powerful feature for texture classification. It has further been determined that when LBP is combined with histograms of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets.

Using the LBP combined with histograms we can represent the face images with a simple data vector.

As LBP is a visual descriptor it can also be used for face recognition tasks, as can be seen in the following step-by-step explanation.

Step-by-Step

Now that we know a little more about face recognition and the LBPH, let’s go further and see the steps of the algorithm:

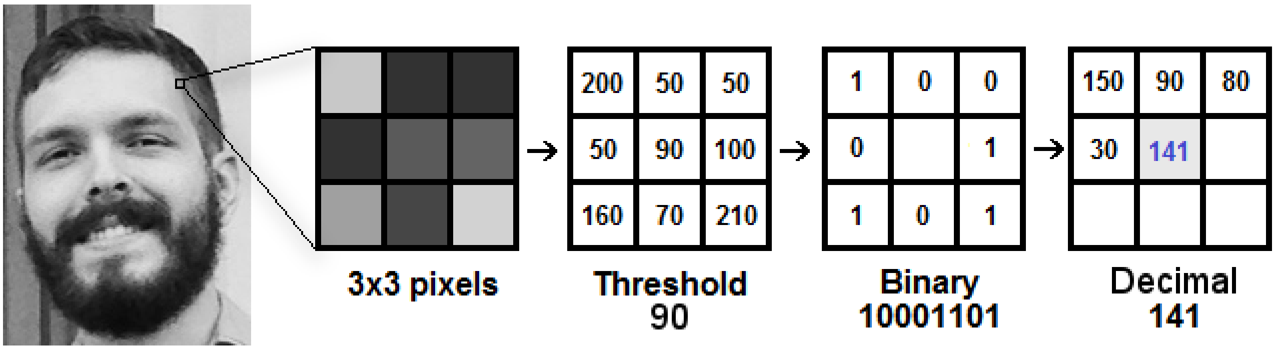
1. **Parameters**: the LBPH uses 4 parameters:

- **Radius**: the radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.
- **Neighbors**: the number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.
- **Grid X**: the number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.
- **Grid Y**: the number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

2. Training the Algorithm: First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize. We need to also set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give you an output. Images of the same person must have the same ID. With the training set already constructed, let’s see the LBPH computational steps.

3. Applying the LBP operation: The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameters **radius** and **neighbors**.

The image below shows this procedure:



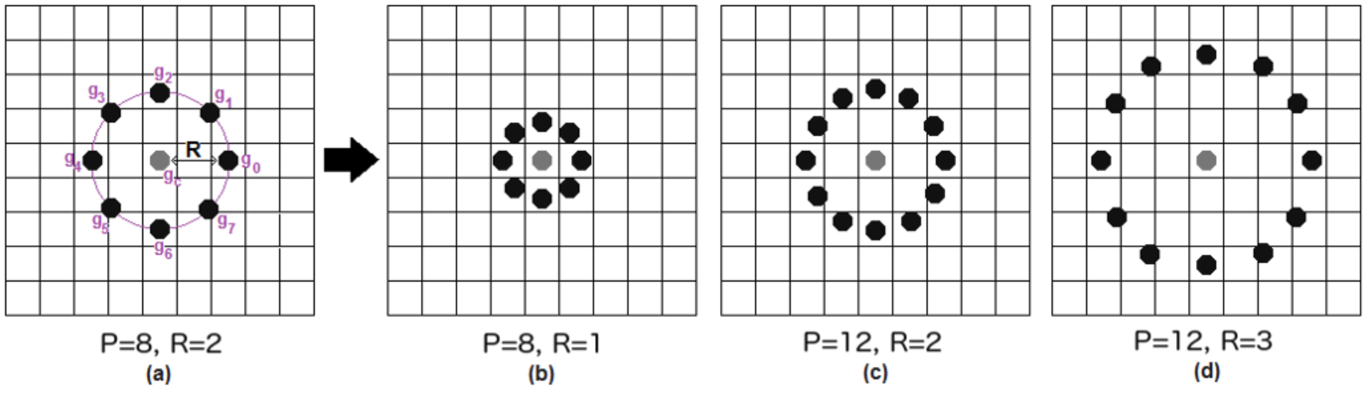
200	50	50
50	90	100
160	70	210

1	0	0
0		1
1	0	1

150	90	80
30	141	

Based on the image above, let’s break it into several small steps so we can understand it easily:

- Suppose we have a facial image in grayscale.
- We can get part of this image as a window of 3x3 pixels.
- It can also be represented as a 3x3 matrix containing the intensity of each pixel (0~255).
- Then, we need to take the central value of the matrix to be used as the threshold.
- This value will be used to define the new values from the 8 neighbors.
- For each neighbor of the central value (threshold), we set a new binary value. We set 1 for values equal or higher than the threshold and 0 for values lower than the threshold.
- Now, the matrix will contain only binary values (ignoring the central value). We need to concatenate each binary value from each position from the matrix line by line into a new binary value (e.g. 10001101). Note: some authors use other approaches to concatenate the binary values (e.g. clockwise direction), but the final result will be the same.
- Then, we convert this binary value to a decimal value and set it to the central value of the matrix, which is actually a pixel from the original image.
- At the end of this procedure (LBP procedure), we have a new image which represents better the characteristics of the original image.
- **Note**: The LBP procedure was expanded to use a different number of radius and neighbors, it is called Circular LBP.



P=8, R=2
(a)

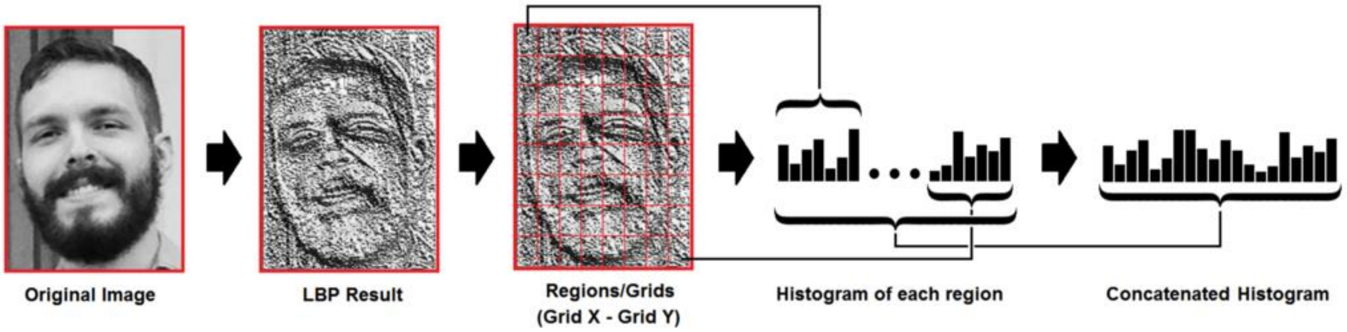
P=8, R=1
(b)

P=12, R=2
(c)

P=12, R=3
(d)

It can be done by using **bilinear interpolation**. If some data point is between the pixels, it uses the values from the 4 nearest pixels (2x2) to estimate the value of the new data point.

4. Extracting the Histograms: Now, using the image generated in the last step, we can use the **Grid X** and **Grid Y** parameters to divide the image into multiple grids, as can be seen in the following image:



Based on the image above, we can extract the histogram of each region as follows:

- As we have an image in grayscale, each histogram (from each grid) will contain only 256 positions (0~255) representing the occurrences of each pixel intensity.
- Then, we need to concatenate each histogram to create a new and bigger histogram. Supposing we have 8x8 grids, we will have 8x8x256=16.384 positions in the final histogram. The final histogram represents the characteristics of the image original image.

The LBPH algorithm is pretty much it.

5. Performing the face recognition: In this step, the algorithm is already trained. Each histogram created is used to represent each image from the training dataset. So, given an input image, we perform the steps again for this new image and creates a histogram which represents the image.

- So to find the image that matches the input image we just need to compare two histograms and return the image with the closest histogram.
- We can use various approaches to compare the histograms (calculate the distance between two histograms), for example: **euclidean distance**, **chi-square**, **absolute value**, etc. In this example, we can use the Euclidean distance (which is quite known) based on the following formula:

$$D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}$$

- So the algorithm output is the ID from the image with the closest histogram. The algorithm should also return the calculated distance, which can be used as a ‘**confidence**’ measurement. **Note**: don’t be fooled about the ‘confidence’ name, as lower confidences are better because it means the distance between the two histograms is closer.

- We can then use a threshold and the ‘confidence’ to automatically estimate if the algorithm has correctly recognized the image. We can assume that the algorithm has successfully recognized if the confidence is lower than the threshold defined.

Conclusions

- LBPH is one of the easiest face recognition algorithms.

- It can represent local features in the images.

- It is possible to get great results (mainly in a controlled environment).

- It is robust against monotonic gray scale transformations.

- It is provided by the OpenCV library (Open Source Computer Vision Library).

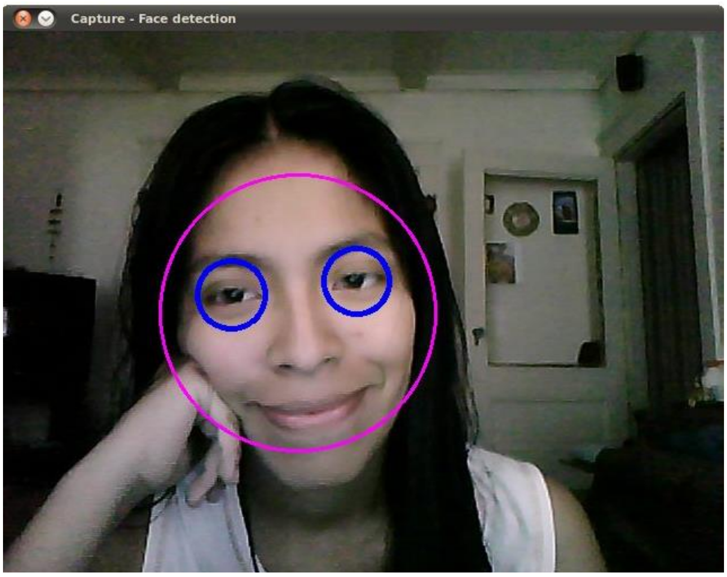
Haar-cascade Detection in OpenCV

OpenCV provides a training method (see **Cascade Classifier Training**) or pretrained models, that can be read using the **cv::CascadeClassifier::load** method. The pretrained models are located in the data folder in the OpenCV installation or can be found here.

The following code example will use pretrained Haar cascade models to detect faces and eyes in an image. First, a **cv::CascadeClassifier** is created and the necessary XML file is loaded using the **cv::CascadeClassifier::load** method. Afterwards, the detection is done using the **cv::CascadeClassifier::detectMultiScale** method, which returns boundary rectangles for the detected faces or eyes.

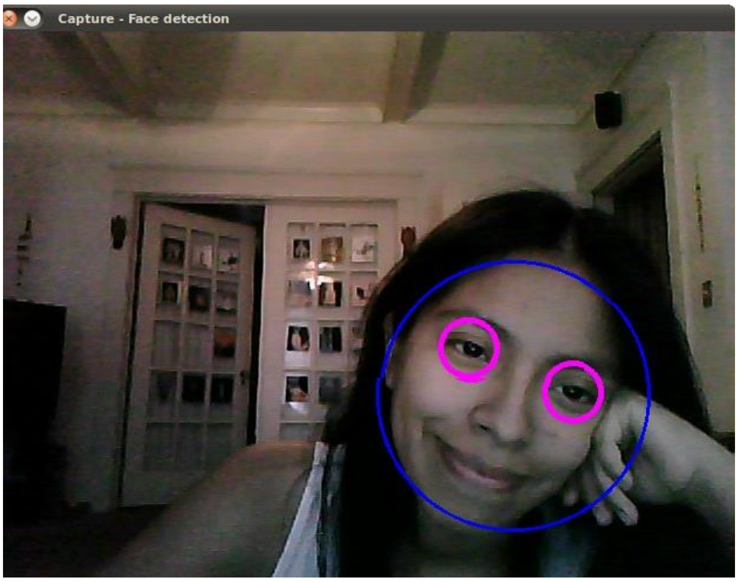
Result:

1.Here is the result of running the code and using as input the video stream of a built-in webcam:



Be sure the program will find the path of files haarcascade_frontalface_alt.xml and haarcascade_eye_tree_eyeglasses.xml. They are located in opencv/data/haarcascades

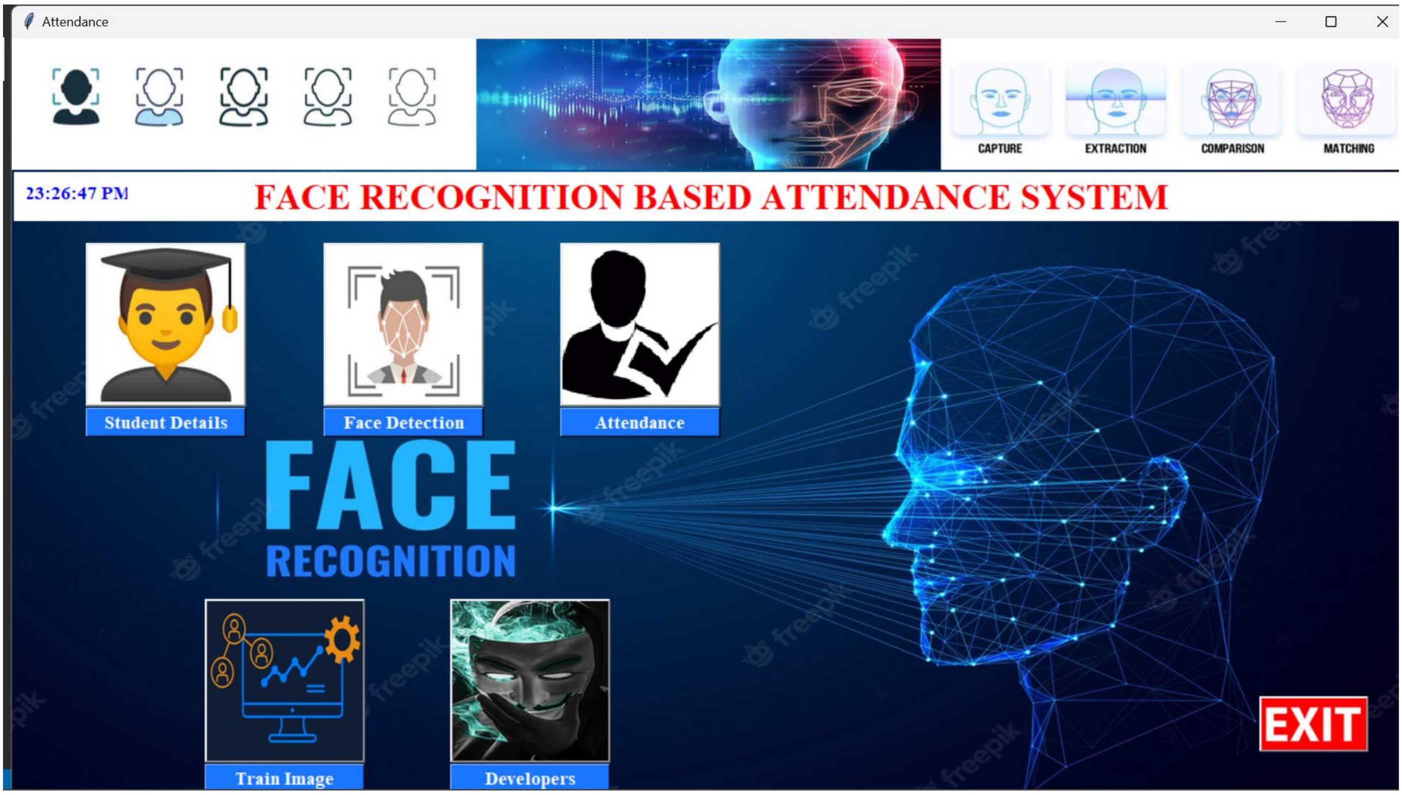
2.This is the result of using the file lbpcascade_frontalface.xml(LBP trained) for the face detection.For the eyes we keep using the file used in the tutorial



For further more information, refer the opencv documentation for face detection https://docs.opencv.org/3.4/da/d60/tutorial_face_main.html .

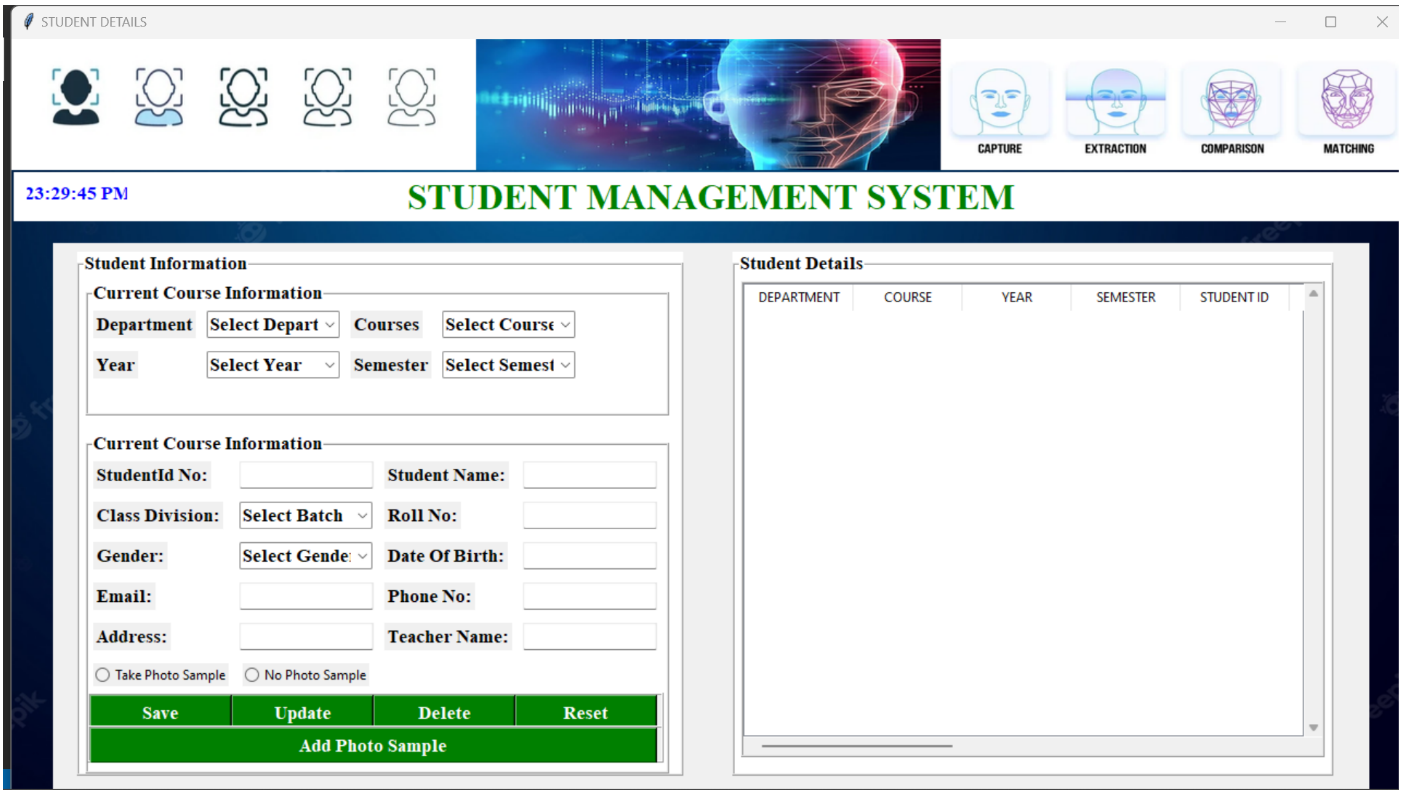
4. EXPERIMENTAL RESULTS:

This is the home page of our application which consists the 6 buttons they are Student Details , Face Detection , Attendance , Train Image , Developers , Exit



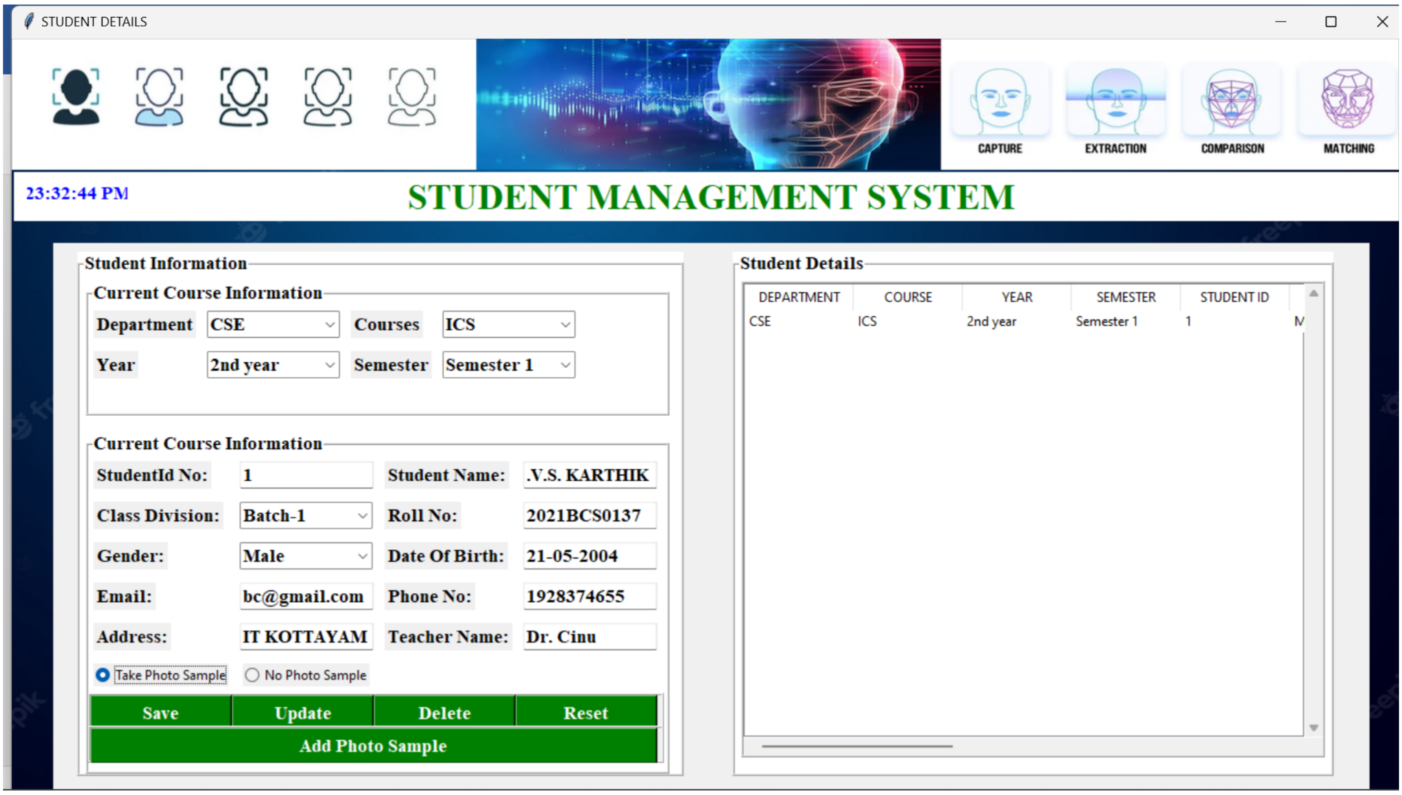
1.Student Details:

This is the home page of student details.



Here we can enter the details and save them to the databse.

Here we can also update and delete our details.



23:32:44 PM

STUDENT MANAGEMENT SYSTEM

Student Information

Current Course Information

Department	CSE	Courses	ICS
Year	2nd year	Semester	Semester 1

Current Course Information

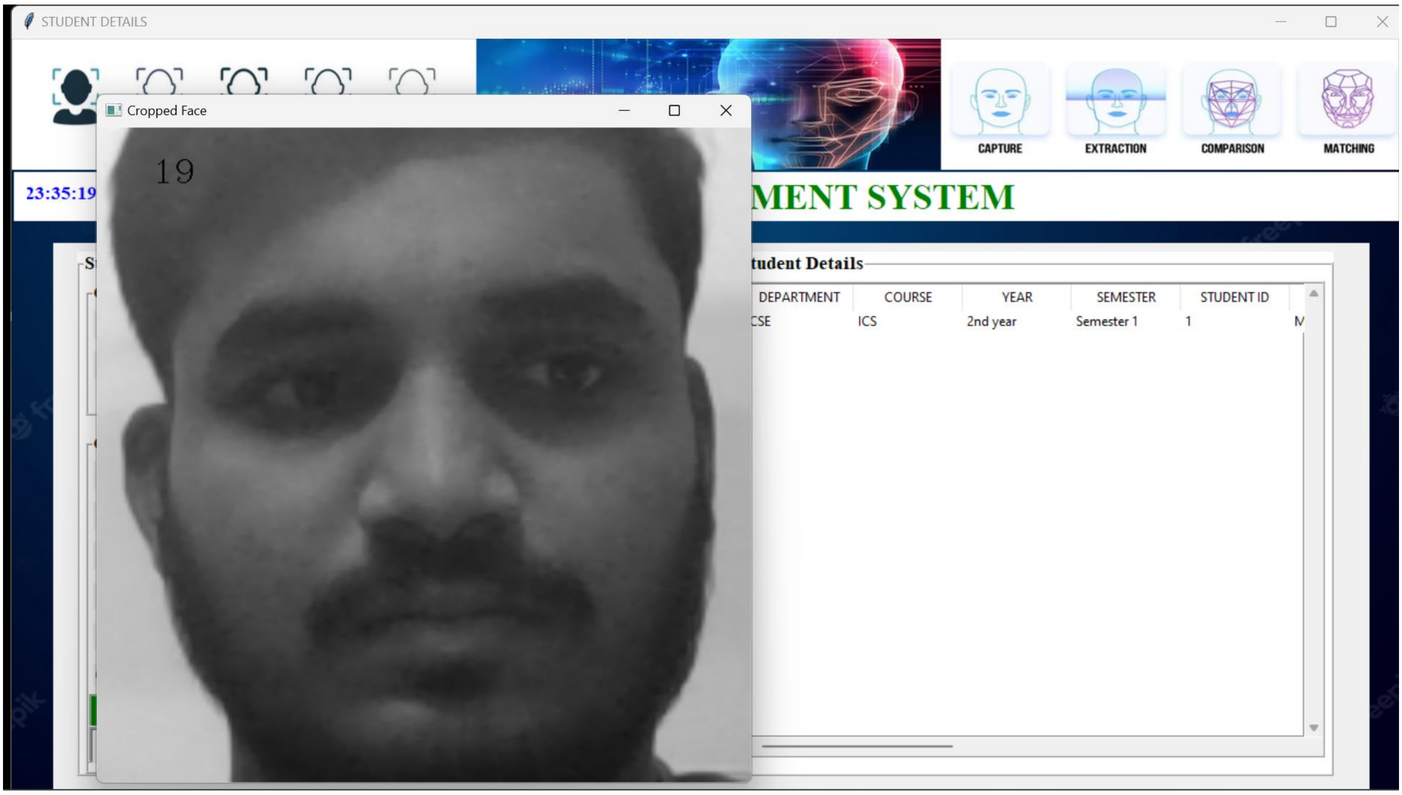
StudentId No:	1	Student Name:	.V.S. KARTHIK
Class Division:	Batch-1	Roll No:	2021BCS0137
Gender:	Male	Date Of Birth:	21-05-2004
Email:	bc@gmail.com	Phone No:	1928374655
Address:	IT KOTTAYAM	Teacher Name:	Dr. Cinu

● Take Photo Sample ○ No Photo Sample

Student Details

DEPARTMENT	COURSE	YEAR	SEMESTER	STUDENT ID	
CSE	ICS	2nd year	Semester 1	1	M

We add the photo samples(100) of the corresponding student by clikcking Add photo sample button



23:35:19

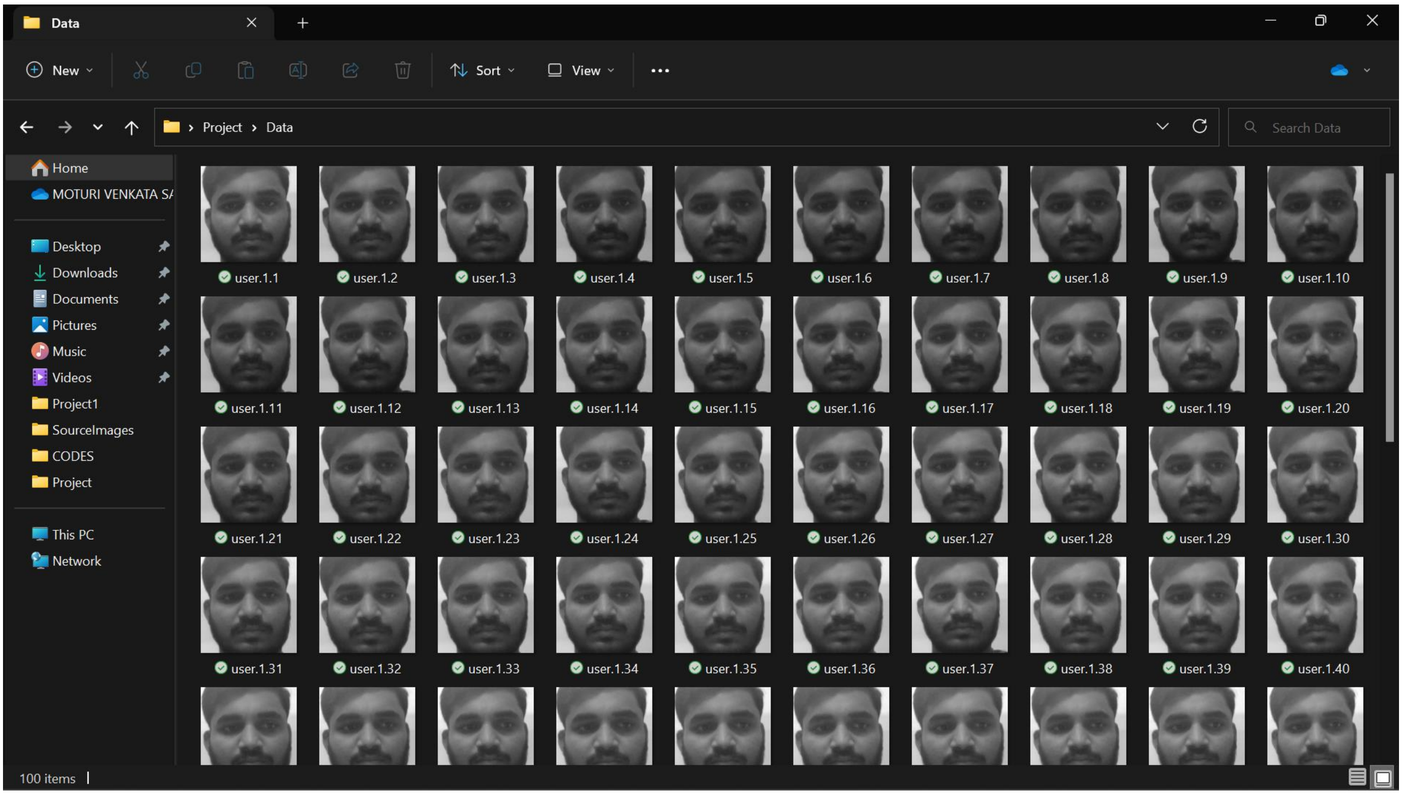
MENT SYSTEM

tudent Details

DEPARTMENT	COURSE	YEAR	SEMESTER	STUDENT ID	
CSE	ICS	2nd year	Semester 1	1	M

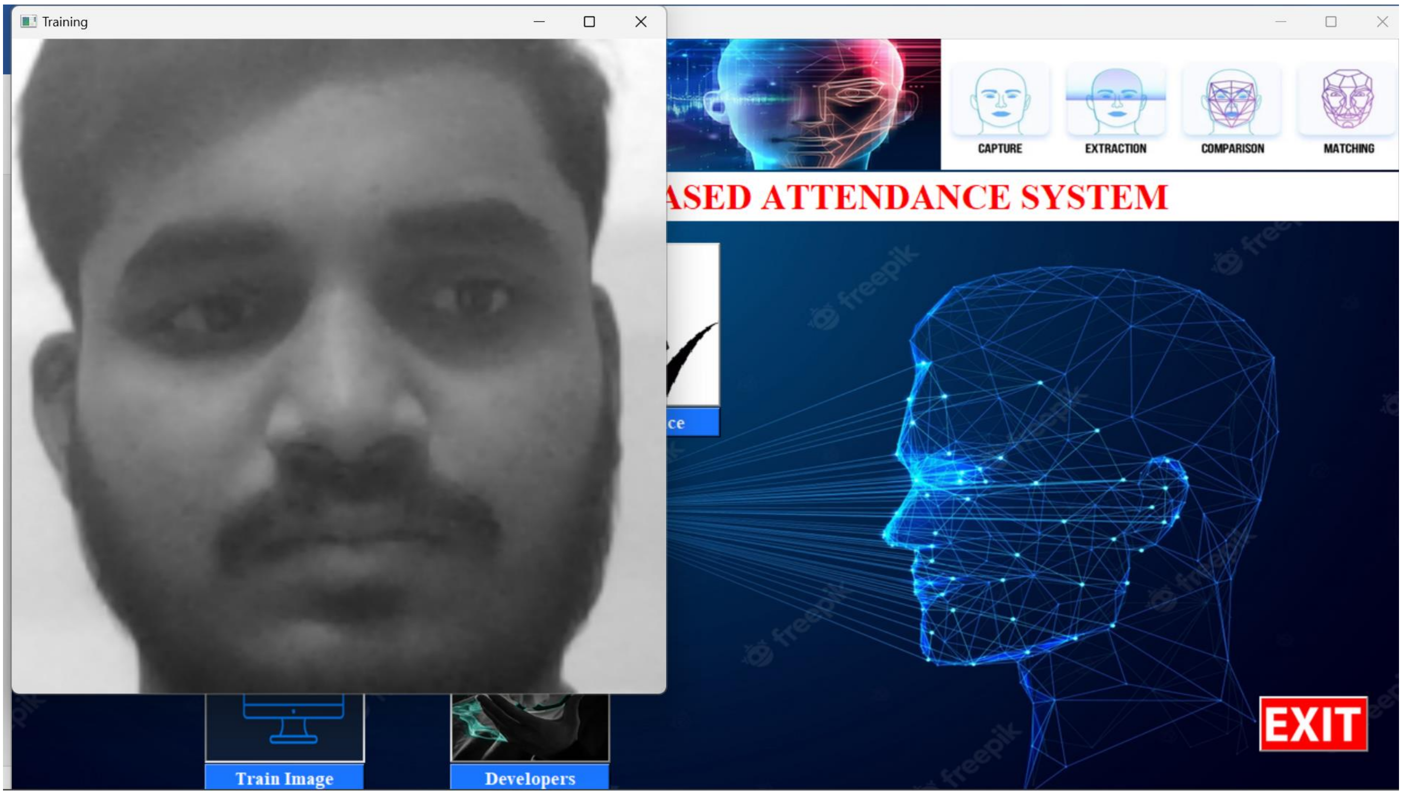
The captured photos of the corresponding student will be automatically stored in a folder named Data

The name of the image captured will be of format user.{user_id}.{image_id}.



2.Train Image:

The images captured will be trained by clicking Train Image button which works on LBPH Algorithm. The trained data will be automatically get stored into classifier.xml

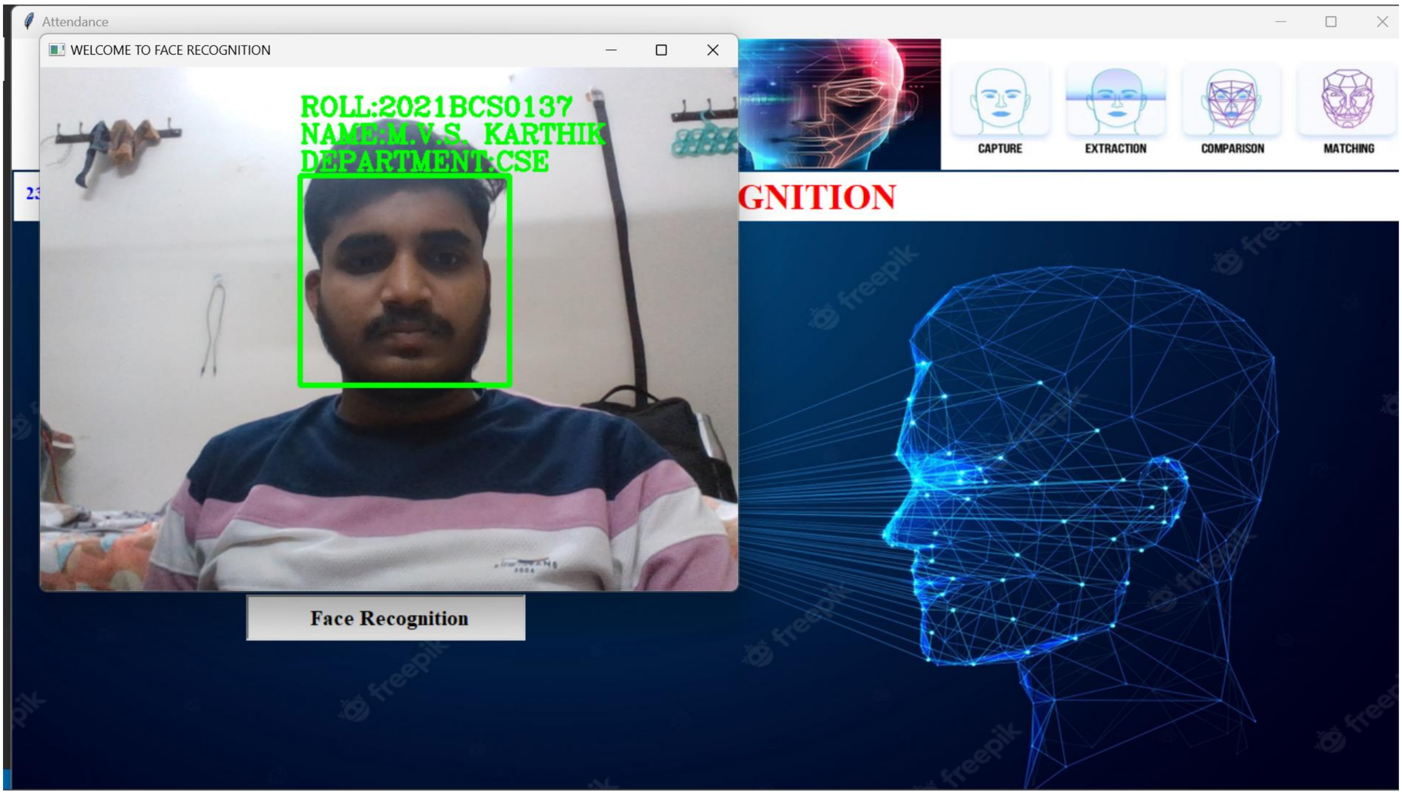


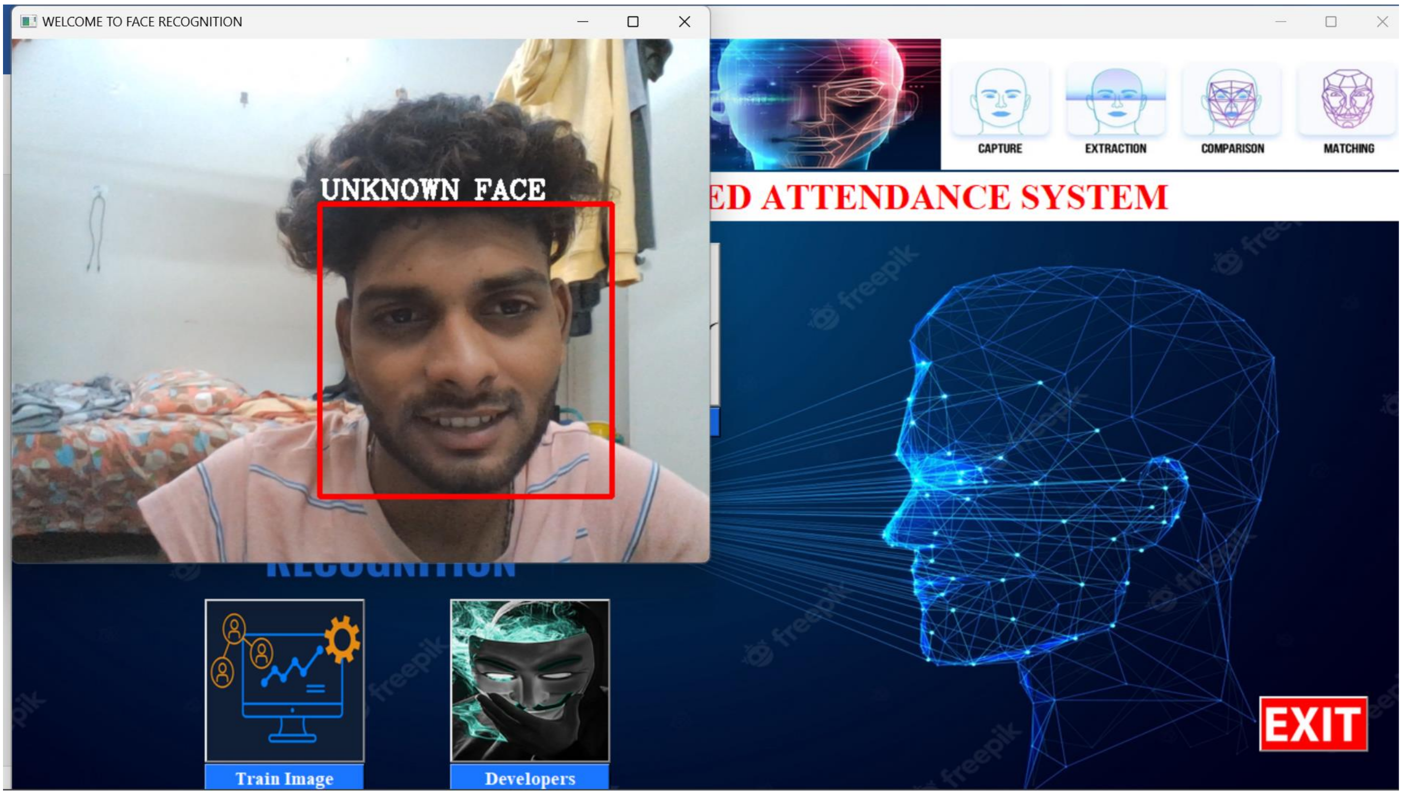
3.Face Recognition:

By clicking the face recognition button the camera detects the face of the student with the images that are trained.

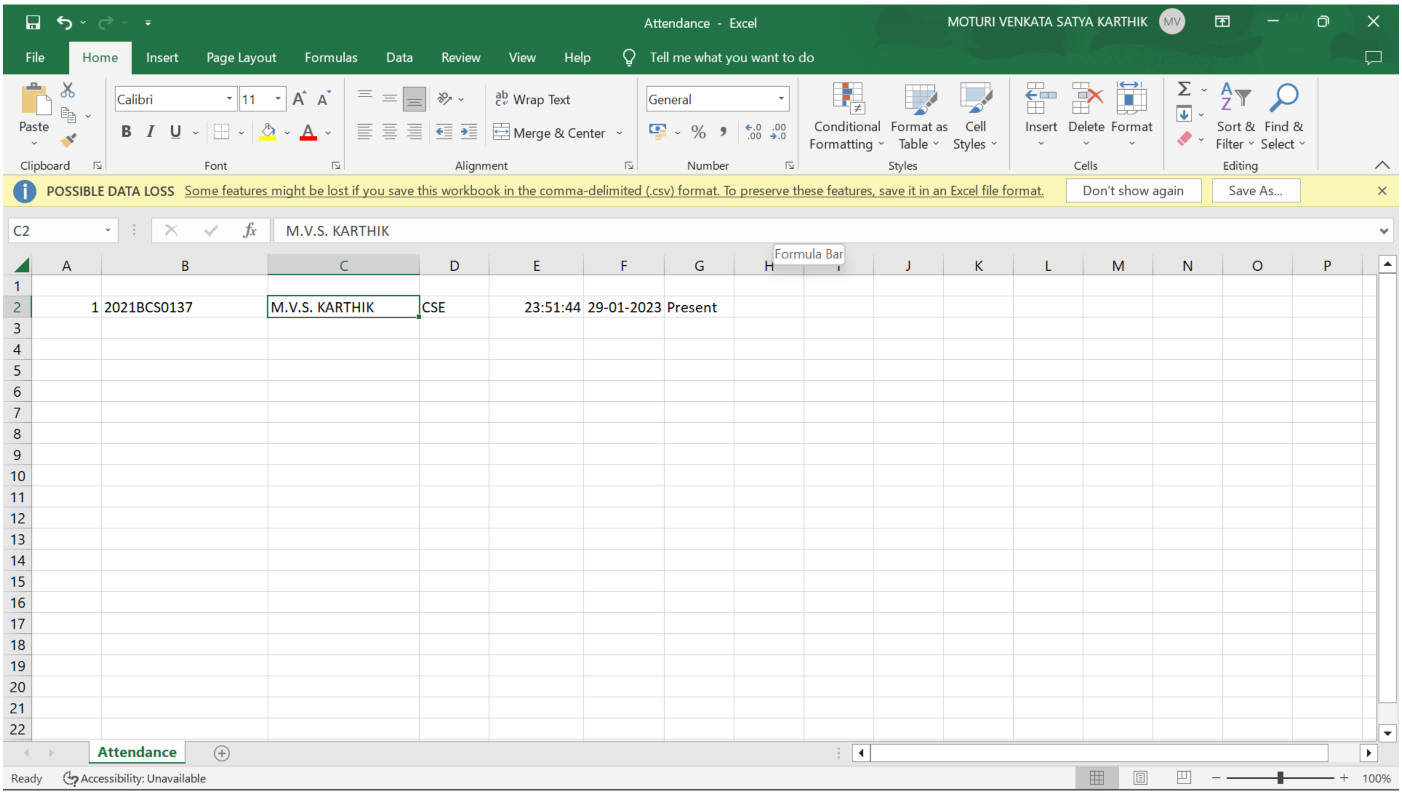
If the face gets detected the attendance of the corresponding student will be automatically gets stored in attendance.csv file.

It shows unknown face if it doesn't get match with the trained images.





The attendance report of the detected face of the student is saved as below:



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format. Don't show again Save As...

C2 M.V.S. KARTHIK

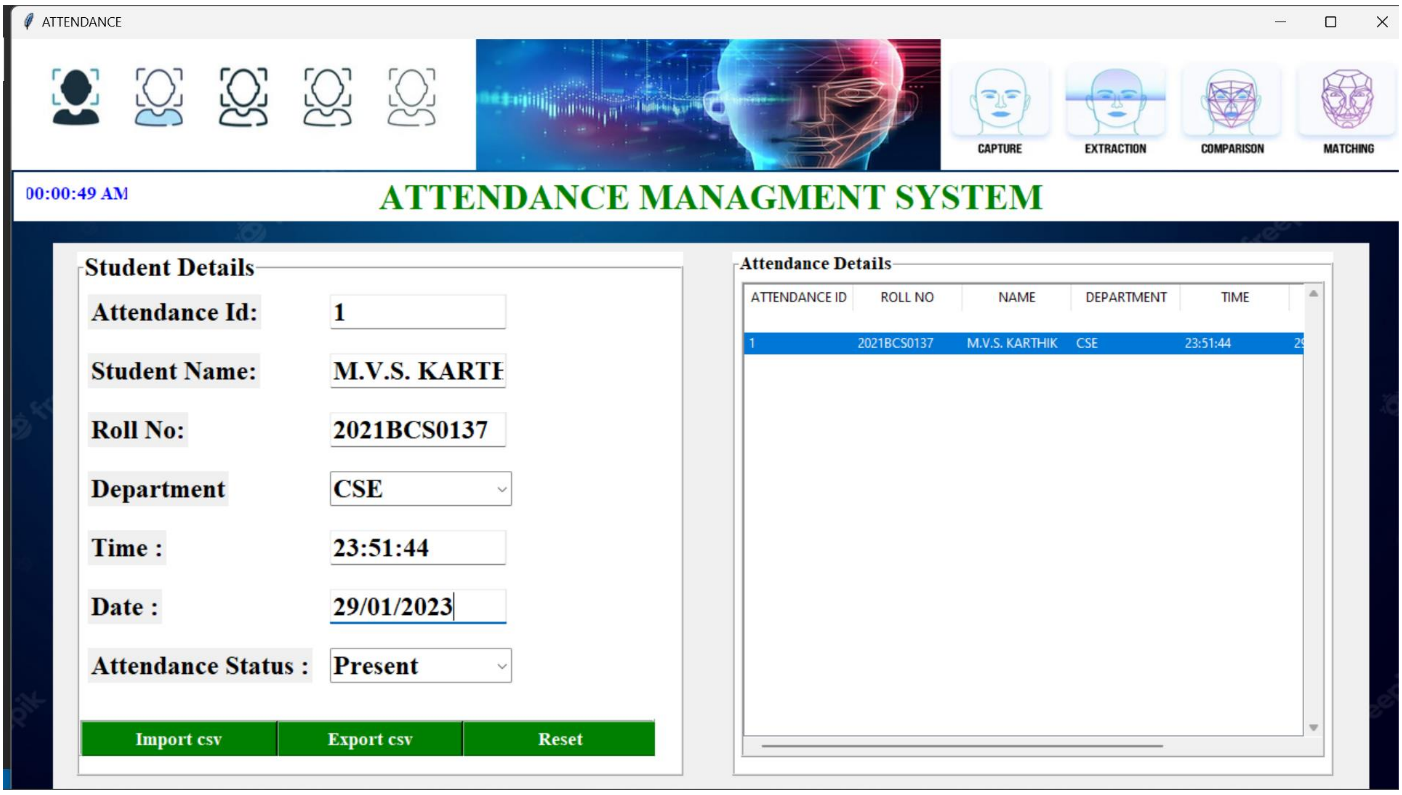
	A	B	C	D	E	F	G
1							
2	1	2021BCS0137	M.V.S. KARTHIK	CSE	23:51:44	29-01-2023	Present

4.Attendance:

Here we can view the saved attendance of the students.

When we click the import csv button the attendance stored in attendance.csv will be displayed here.

We can export our attendance into another file using export csv button.



00:00:49 AM

ATTENDANCE MANAGMENT SYSTEM

Student Details

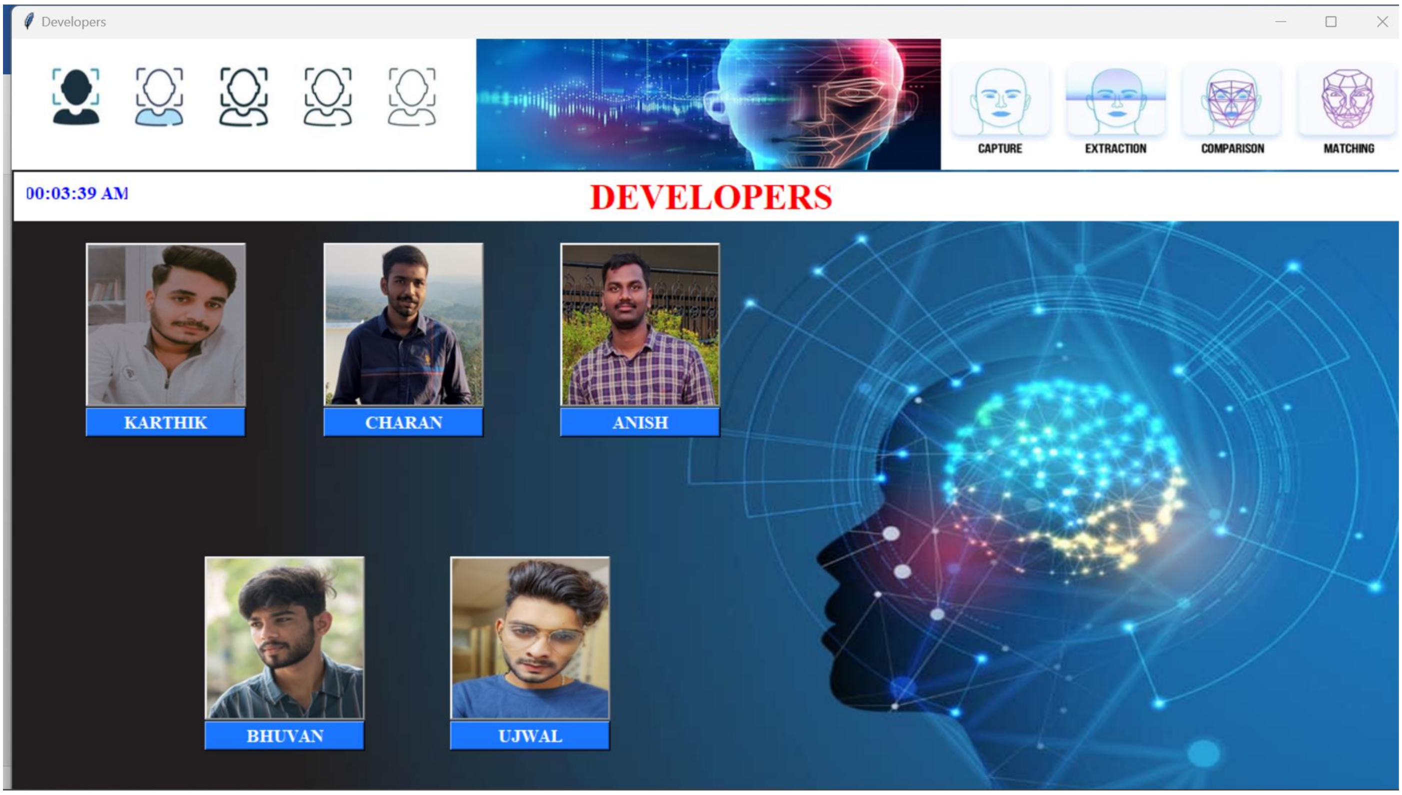
Attendance Id:	1
Student Name:	M.V.S. KARTH
Roll No:	2021BCS0137
Department	CSE
Time :	23:51:44
Date :	29/01/2023
Attendance Status :	Present

Attendance Details

ATTENDANCE ID	ROLL NO	NAME	DEPARTMENT	TIME
1	2021BCS0137	M.V.S. KARTHIK	CSE	23:51:44

5.Developers:

Here we can view the photos of the developers.



5. CONCLUSION:

Face recognition based attendance system can provide a convenient and effective solution for automating attendance tracking, but it is important to consider the potential risks and ensure that the system is implemented responsibly.

Face recognition systems are part of facial image processing applications and their significance as a research area are increasing recently. Implementations of system are crime prevention, video surveillance, person verification, and similar security activities. The face recognition system implementation can be part of Universities.

Face Recognition Based Attendance System has been envisioned for the purpose of reducing the errors that occur in the traditional (manual) attendance taking system. The aim is to automate and make a system that is useful to the organization such as an institute.

The efficient and accurate method of attendance in the office environment that can replace the old manual methods. This method is secure enough, reliable and available for use. Proposed algorithm is capable of detect multiple faces, and performance of system has acceptable good results.

TEAM DETAIL’S:

Team Number: 7

ROLL NUMBER	**NAME**	**EMAIL**
2021BCS0137	M V S KARTHIK	moturi21bcs137@iiitkottayam.ac.in
2021BCS0099	K ANISH REDDY	konyala21bcs99@iiitkottayam.ac.in
2021BCS0091	V CHARAN SAI	vattikuti21bcs91@iiitkottayam.ac.in
2021BCS0153	JADHAV BHUVAN SAI	jadhav21bcs153@iiitkottayam.ac.in
2021BCS0187	P UJWAL KUMAR	perla21bcs187@iiitkottayam.ac.in