**Mini Project Report**

**On**

**“Face Recognition Based Attendance System”**

Submitted in partial fulfilment for the award of the degree of

**Bachelor of Technology**

**in**

**Information Technology**

**By**

**N Hiranmayee 21r1a1241**

**p shiva krishna 21r21a1245**

**s pranav 21r21a1253**

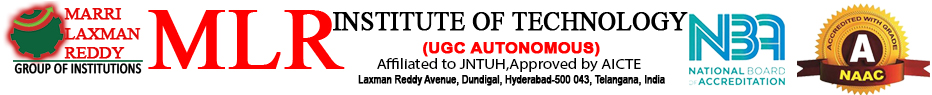
**t sai kumar 21r21a1255**

Under the guidance of

**M Harshini**

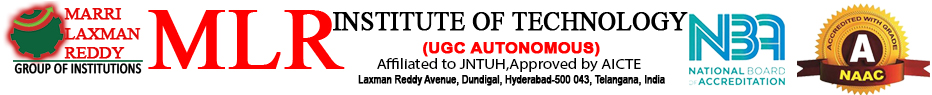
**Assistant Professor**

**Department of Information Technology**

******

**2022-23**

**Department of Information Technology**

****

**Department of Information Technology**

**CERTIFICATE**

This is to certify that the project entitled **“Face Recognition Based Attendance System”** has been submitted by **N.Hiranmayee (21R21A1241), p.Shiva Krishna(21R21A1245),**

**S.Pranav(21R21A1253), T.Sai Kumar(21R21A1255)** in the partial fulfilment of the requirements for the award of degree of Bachelor of Technology in Information Technology from Jawaharlal Nehru Technological University, Hyderabad. The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

**Internal Guide Head of the Department**

**M Harshini**

**Assistant Professor Dr.N.V.Rajasekhar Reddy**

**External Examiner**

**DECLARATION**

I hereby declare that the project entitled **“Face Recognition Based Attendance System”** is the work done during the period from **October 2022 to February 2023** and is submitted in the partial fulfilment of the requirements for the award of degree of Bachelor of technology in Information Technology from Jawaharlal Nehru Technology University, Hyderabad. The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

**N.Hiranmayee(21R21A1241)**

**P.Shiva Krishna(21R21A1245)**

**S.Pranav(1R21A1253)**

**T.Sai Kumar(21R21A1255)**

**ACKNOWLEDGEMENT**

There are many people who helped me directly and indirectly to complete my project successfully. I would like to take this opportunity to thank one and all.

First of all I would like to express my deep gratitude towards my internal guide **M.Harshini Assistant Professor, Department of IT** for her support in the completion of my dissertation. I wish to express my sincere thanks to **Dr.N.V.RAJASEKHAR REDDY, HOD, Department of IT** and also to Principal **Dr.K.SRINIVASA RAO** for providing the facilities to complete the dissertation.

I would like to thank all our faculty, coordinators and friends for their help and constructive criticism during the project period. Finally, I am very much indebted to our parents for their moral support and encouragement to achieve goals.

**N.Hiranmayee(21R21A1241)**

**P.Shiva Krishna(21R21A1245)**

**S.Pranav(1R21A1253)**

**T.Sai Kumar(21R21A1255)**

**TABLE OF CONTENTS**

|  |
| --- |
| 1. **Introduction 1**    1. Existing System    2. Proposed System 2. **System Requirements 2**    1. Hardware Requirements    2. Software Requirements 3. **System Design 3-4**    1. Description of Modules    2. Description of Algorithms       1. Local Binary Patterns Histogram Algorithm       2. Haar Cascade Algorithm 4. **Implementation 5-11**    1. Installation of Visual Studio    2. Installation of Open CV    3. Coding       1. Code for Create dataset       2. Code for train dataset       3. Code for Recognition       4. Code for XL write 5. **Results and Discussion 12-13**    1. Dataset creation    2. Verification    3. Output 6. **Future Scope 14** 7. **Conclusion 15** 8. **References 16** |

**ABSTRACT**

A face recognition attendance system automatically identifies and confirms a person and records attendance based on their face detection. Unlike other forms of biometric technology, such as fingerprint recognition, which captures identity by touching, a facial identification system manages without the approach of direct contact. Face recognition records the attendance with marking time. This application identifies and verifies a person by mapping his or her facial features with the trained data. The machine will be trained well before with the person’s id and face so that it identifies accurately when implemented.

**1.INTRODUCTION**

This application identifies and verifies a person by mapping his or her facial features with the trained data. The machine will be trained well before with the person’s id and face so that it identifies accurately when implemented. To identify faces the local features method splits the face into a number of local features and then identifies the face using these features.

**1.1 EXISTING SYSTEM**

**Fingerprint Based recognition system & Manual Attendance**

In the Fingerprint based existing attendance system, a portable fingerprint device need to be configured with the students fingerprint earlier. Later either during the lecture hours or before, the student need store the finger print on the configured device to ensure their attendance for the day. In manual attendance, the teachers has to call out the students names or roll numbers for marking their attendance.

**1.2 PROPOSED SYSTEM**

The task of the proposed system is to capture the face of each student and to store it in the database for their attendance. This application identifies and verifies a person by mapping his or her facial features with the trained data. Initially we create the dataset, later we train the data and check with the recognition.

**2.SYSTEM REQUIREMENTS**

* 1. **HARDWARE REQUIREMENTS**

RAM : 4 GB+

Input Devices : Web Cam

Output De vices : PC screen

**2.2** **SOFTWARE REQUIREMENTS**

Operating System : Windows 10

Programming Language : Python

IDE/Workbench : Visual Studio Code

**3.SYSTEM DESIGN**

**3.1 DESCRIPTION OF MODULES**

Modules present in this Face Recognition based attendance system project are:

• Create Dataset Module

• Train Dataset Module

• Recognition Module

• XL Write Module

1. **Create Dataset module:** In this module, it creates its own data set by capturing the user images with the specified identification number.

2. **Train Dataset module:** Here, the previously stored dataset will be trained to the machine. It maps the user ids to the user faces.

3. **Recognition Module:** When the data is successfully trained, we check wheather it recognizes the faces correctly. If it recognizes the attendance will be marked in the XL sheet else if gives a message as unrecognized face.

4. **XL write:** When the face is recognized correctly, attendance will be marked at the specific id. The excel sheet will be automatically created with the file name as respective day date.

**3.2 DESCRIPTION OF ALGORITHMS**

**3.2.1 Local Binary Patterns Histogram Algorithm**

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighbourhood 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.

**3.2.2 Haar Cascade Algorithm**

Haar cascade is an algorithm that can detect objects in images, irrespective of their scale in image and location. This algorithm is not so complex and can run in real-time. We can train a haar-cascade detector to detect various objects like cars, bikes, buildings, fruits, etc. Haar cascade uses the cascading window, and it tries to compute features in every window and classify whether it could be an object.

Haar cascade works as a classifier. It classifies positive data points → that are part of our detected object and negative data points → that don’t contain our object.

• Haar cascades are fast and can work well in real-time.

• Haar cascade is not as accurate as modern object detection techniques are.

• Haar cascade has a downside. It predicts many false positives.

• Simple to implement, less computing power required.

In OpenCV library manages a repository containing all popular haar cascades that can be used for:

• Human face detection

• Eye detection

• Nose / Mouth detection

• Vehicle detection

**ARCHITECTURE**

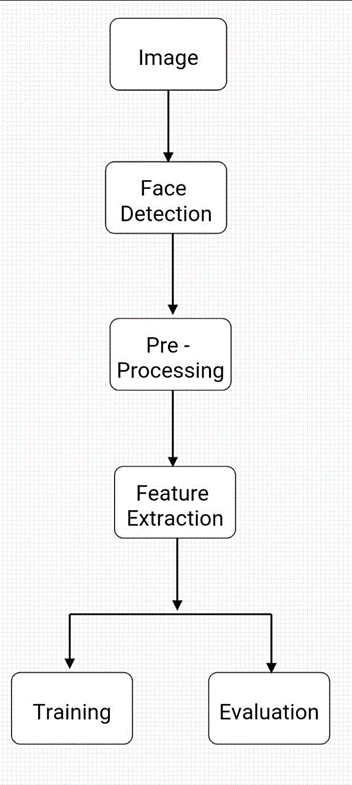
****

Fig: Architecture diagram

**4.IMPLEMENTATION**

**4.1 Installation of Visual Studio**

Step 1 - Make sure your computer is ready for Visual Studio.

Step 2 - Download Visual Studio.

Step 3 - Install the Visual Studio Installer.

Step 4 - Choose workloads.

Step 5 - Choose individual components (optional)

Step 6 - Install language packs (optional)

Step 7 - Select the install location Step 8 - Start developing

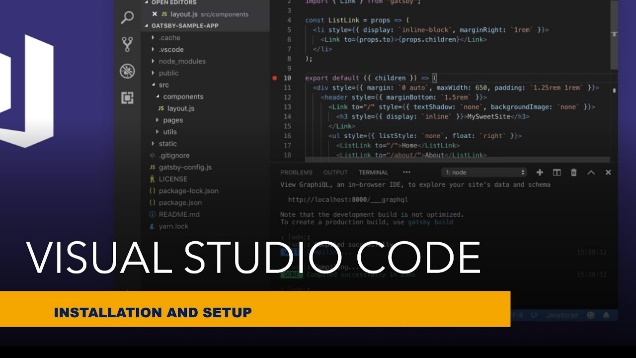


Fig:4.1 Visual Studio Code

**4.2 Installation of Open CV**

Installation of open cv is easy if pip is already installed. PIP is a package management system used to install and manage software packages/libraries written in Python. These files are stored in a large “on-line repository” termed as Python Package Index (PyPI).

To check weather pip is already installed or not run the following command in the command prompt.

****

Fig:4.2 Checking for pip

To install open cv in our system:

Command:

pip install opencv

The above command installs open cv in our system

To run python files:

Command:

<path of the file><name of python files>

Example:

C:/Users/rishi/Desktop/facereg/createdataset.py

**4.3 Coding**

**4.3.1 Code For Create Data set**

import cv2

import os

def assure\_path\_exists(path):

    dir = os.path.dirname(path)

    if not os.path.exists(dir):

        os.makedirs(dir)

face\_id = input('Enter your ID ')

vid\_cam = cv2.VideoCapture(0)

face\_detector = cv2.CascadeClassifier('c:/users/rishi/appdata/local/programs/python/python311/lib/site-packages/cv2/data/haarcascade\_frontalface\_default.xml')

count = 0

assure\_path\_exists("C:/Users/rishi/OneDrive/Desktop/Face\_Recognition\_Attendance\_System/Create data")

while (True):

    \_, image\_frame = vid\_cam.read()

    gray = cv2.cvtColor(image\_frame, cv2.COLOR\_BGR2GRAY)

    faces = face\_detector.detectMultiScale(gray, 1.3, 5)

    for (x, y, w, h) in faces:

        cv2.rectangle(image\_frame, (x, y), (x + w, y + h), (255, 0, 0), 2)

        count += 1

        cv2.imwrite("C:/Users/rishi/OneDrive/Desktop/Face\_Recognition\_Attendance\_System/Create data/User." + str(face\_id) + '.' + str(count) + ".jpg", gray[y:y + h, x:x + w])

        cv2.imshow('frame', image\_frame)

    if cv2.waitKey(100) & 0xFF == ord('q'):

        break

    elif count >= 50:

        print("Successfully Captured")

        break

vid\_cam.release()

cv2.destroyAllWindows()

**4.3.2 Coding for Train Data Set**

import os,cv2;

import numpy as np

from PIL import Image;

recognizer = cv2.face.LBPHFaceRecognizer\_create()

detector= cv2.CascadeClassifier("c:/users/rishi/appdata/local/programs/python/python311/lib/site-packages/cv2/data/haarcascade\_frontalface\_default.xml");

def getImagesAndLabels(path):

    imagePaths=[os.path.join(path,f) for f in os.listdir(path)]

    faceSamples=[]

    Ids=[]

    for imagePath in imagePaths:

        pilImage=Image.open(imagePath).convert('L')

        imageNp=np.array(pilImage,'uint8')

        Id=int(os.path.split(imagePath)[-1].split(".")[1])

        faces=detector.detectMultiScale(imageNp)

        for (x,y,w,h) in faces:

            faceSamples.append(imageNp[y:y+h,x:x+w])

            Ids.append(Id)

    return faceSamples,Ids

faces,Ids = getImagesAndLabels('C:/Users/rishi/OneDrive/Desktop/Face\_Recognition\_Attendance\_System/Create data')

s = recognizer.train(faces, np.array(Ids))

print("Successfully trained")

recognizer.write('C:/Users/rishi/OneDrive/Desktop/Face\_Recognition\_Attendance\_System/trainer.yml')

**4.3.3 Code For Recognition**

import cv2

import numpy as np;

import xlwrite

#import firebase\_admin

#import firebase as Firebase;

#import firebase.firebase\_ini as fire;

#import firebase

import time

#import sys

from playsound import playsound

start = time.time()

period = 8

face\_cas = cv2.CascadeClassifier('c:/users/rishi/appdata/local/programs/python/python311/lib/site-packages/cv2/data/haarcascade\_frontalface\_default.xml')

cap = cv2.VideoCapture(0);

recognizer = cv2.face.LBPHFaceRecognizer\_create();

recognizer.read('C:/Users/rishi/OneDrive/Desktop/Face\_Recognition\_Attendance\_System/trainer.yml');

flag = 0;

id = 0;

filename = 'filename';

dict = {

    'item1': 1

}

font = cv2.FONT\_HERSHEY\_SIMPLEX

while True:

    ret, img = cap.read();

    gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY);

    faces = face\_cas.detectMultiScale(gray, 1.3, 7);

    for (x, y, w, h) in faces:

        roi\_gray = gray[y:y + h, x:x + w]

        cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2);

        id, conf = recognizer.predict(roi\_gray)

        if (conf < 50):

            if (id == 1241):

                id = 'Hiranmayee'

                if ((str(id)) not in dict):

                    filename = xlwrite.output('attendance', 'class1', 1, id, 'yes');

                    dict[str(id)] = str(id);

           # elif (id == 3):

            #    id = 'Rishitha'

             #   if ((str(id)) not in dict):

              #      filename = xlwrite.output('attendance', 'class1', 2, id, 'yes');

               #     dict[str(id)] = str(id);

            elif (id == 4):

                id = 'Srinivas'

                if ((str(id)) not in dict):

                    filename = xlwrite.output('attendance', 'class1', 3, id, 'yes');

                    dict[str(id)] = str(id);

            elif (id == 1253):

                id = 'Pranav'

                if ((str(id)) not in dict):

                    filename = xlwrite.output('attendance', 'class1', 4, id, 'yes');

                    dict[str(id)] = str(id);

            elif (id == 1255):

                id = 'Sai\_Kumar'

                if ((str(id)) not in dict):

                    filename = xlwrite.output('attendance', 'class1', 3, id, 'yes');

                    dict[str(id)] = str(id);

        else:

            id = 'Unknown, can not recognize'

            flag = flag + 1

            break

        cv2.putText(img, str(id) + " " + str(conf), (x, y - 10), font, 0.55, (120, 255, 120), 1)

    cv2.imshow('frame', img);

    if flag == 10:

        playsound('transactionSound.mp3')

        print("Transaction Blocked")

        break;

    if time.time() > start + period:

        break;

    if cv2.waitKey(100) & 0xFF == ord('q'):

        break;

cap.release();

cv2.destroyAllWindows();

**4.3.4 Code For XL Write**

import xlwt;

from datetime import datetime;

from xlrd import open\_workbook;

from xlwt import Workbook;

from xlutils.copy import copy

from pathlib import Path

def output(filename, sheet,num, name, present):

    my\_file = Path('C:/Users/rishi/OneDrive/Desktop/Face\_Recognition\_Attendance\_System/attendance files/'+filename+str(datetime.now().date())+'.xls');

    if my\_file.is\_file():

        rb = open\_workbook('C:/Users/rishi/OneDrive/Desktop/Face\_Recognition\_Attendance\_System/attendance files/'+filename+str(datetime.now().date())+'.xls');

        book = copy(rb);

        sh = book.get\_sheet(0)

        # file exists

    else:

        book = xlwt.Workbook()

        sh = book.add\_sheet(sheet)

    style0 = xlwt.easyxf('font: name Times New Roman, color-index red, bold on',

                         num\_format\_str='#,##0.00')

    style1 = xlwt.easyxf(num\_format\_str='D-MMM-YY')

    #variables = [x, y, z]

    #x\_desc = 'Display'

    #y\_desc = 'Dominance'

    #z\_desc = 'Test'

    #desc = [x\_desc, y\_desc, z\_desc]

    sh.write(0,0,datetime.now().date(),style1);

    col1\_name = 'Name'

    col2\_name = 'Present'

    sh.write(1,0,col1\_name,style0);

    sh.write(1, 1, col2\_name,style0);

    sh.write(num+1,0,name);

    sh.write(num+1, 1, present);

    #You may need to group the variables together

    #for n, (v\_desc, v) in enumerate(zip(desc, variables)):

    fullname=filename+str(datetime.now().date())+'.xls';

    book.save('C:/Users/rishi/OneDrive/Desktop/Face\_Recognition\_Attendance\_System/attendance files/'+fullname)

    return fullname;

**5.RESULTS AND DISCUSSION**

**5.1 Data Set Creation**

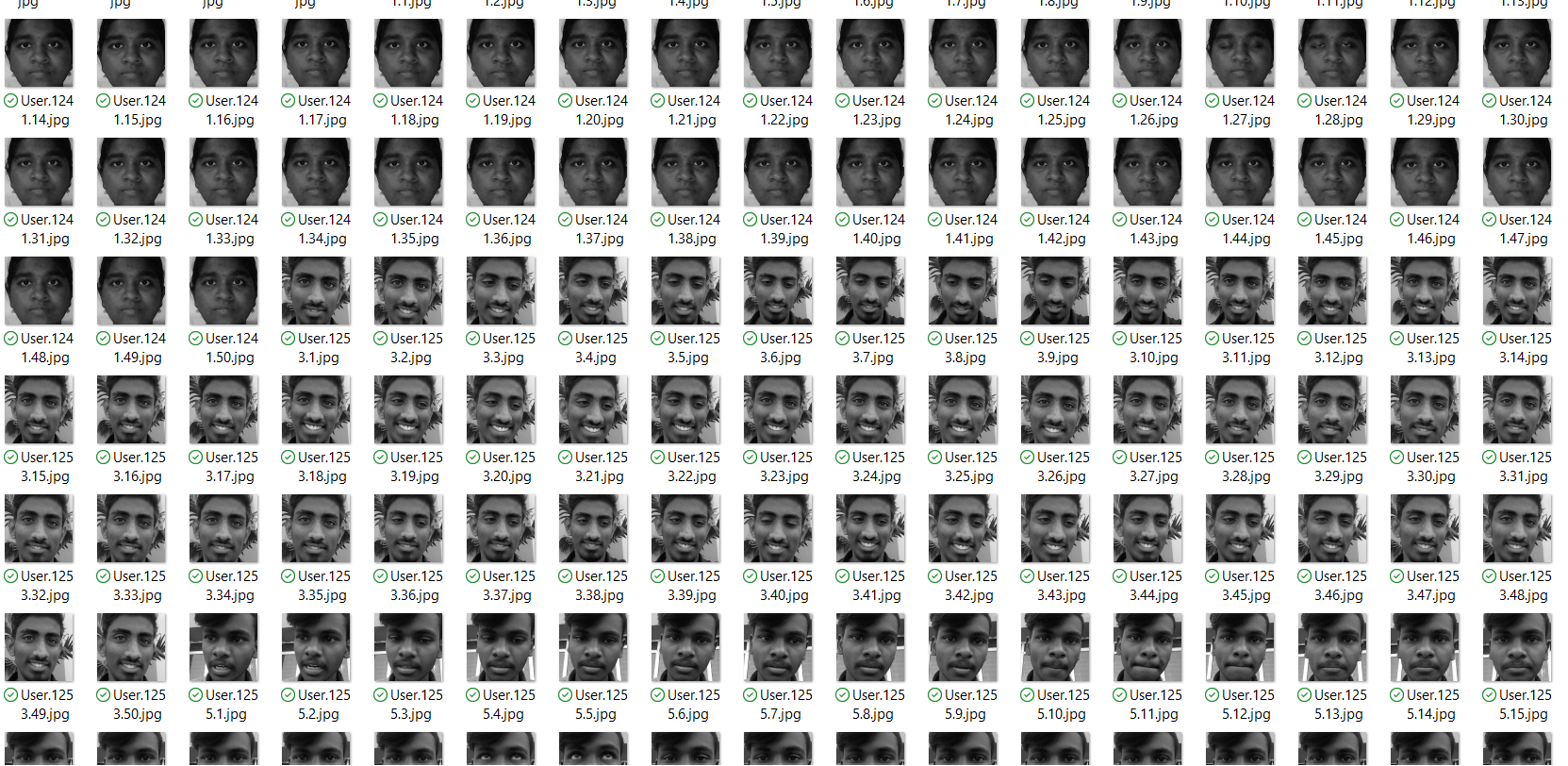
****

Fig:5.1 Dataset creation

**5.2 Verification**

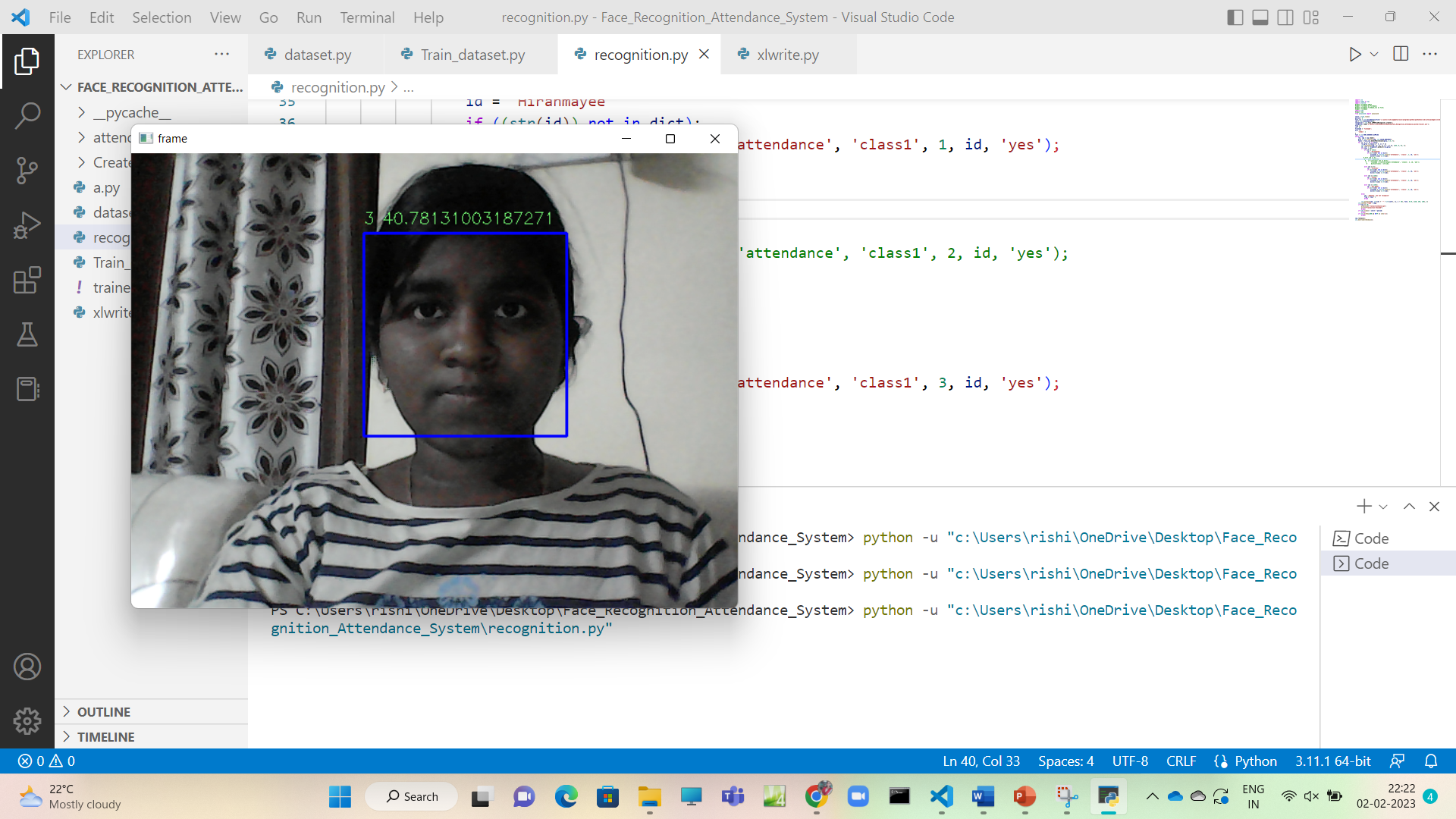


Fig:5.2 Verification with the case

**5.3 Output**

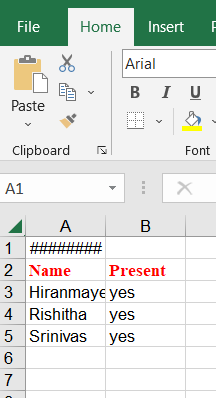
****

Fig:5.3 Result storing in XL Sheet

**6.FUTURE SCOPE**

* This application can be used in classrooms or any place where attendance should be taken.
* This is done easily as no manual work is required.
* Data will be stored in XL sheets which can be accessed easily.
* Can create many profiles and large amount of data sets can also be created.

**7.CONCLUSION**

This system aims to build an effective class attendance system using this technology. The proposed system will be able to mark the attendance via face id. It will detect faces through webcam & then recognize the faces. After recognition, it will mark attendance of the recognised student &update the records in excel sheets.

The solution is both cost-effective and efficient when compared to other biometric solutions. The cost and time saved are even larger because the data acquired from the face recognition-based attendance system is accurate in real life. Because the overall process is automated and human invention is limited.

**8.REFERENCES**

1. N.Sudhakar Reddy, M.V.Sumanth, S.Suresh Babu, "A Counterpart Approach to Attendance and Feedback System using Machine Learning Techniques",Journal of Emerging Technologies and Innovative Research (JETIR), Volume 5, Issue 12, Dec 2018.
2. Dan Wang, Rong Fu, Zuying Luo, "Classroom Attendance Auto-management Based on Deep Learning",Advances in Social Science, Education and Humanities Research, volume 123,ICESAME 2017.
3. Akshara Jadhav, Akshay Jadhav, Tushar Ladhe, Krishna Yeolekar, "AutomateAttendance System Using Face Recognition", International Research Journal of Engineering and Technology (IRJET), Volume 4, Issue 1, Jan 2017.
4. B Prabhavathi, V Tanuja, V Madhu Viswanatham and M Rajashekhara Babu, "A smart technique for attendance system to recognize faces through parallelism", IOP Conf. Series: Materials Science and Engineering 263, 2017.