



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 5, May 2023

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com



Automatic Attendance System using Face Recognition

Om Chincholkar¹, Vedant Fuldeore², Appasha Hulamani³, Arti Bang⁴

U.G. Student, Department of Electronics and Telecommunication Engineering, Vishwakarma Institute of Information Technology College of Engineering, Kondwa, Pune, Maharashtra, India ¹

U.G. Student, Department of Electronics and Telecommunication Engineering, Vishwakarma Institute of Information Technology College of Engineering, Kondwa, Pune, Maharashtra, India ²

U.G. Student, Department of Electronics and Telecommunication Engineering, Vishwakarma Institute of Information Technology College of Engineering, Kondwa, Pune, Maharashtra, India ³

Professor, Department of Electronics and Telecommunication Engineering, Vishwakarma Institute of Information Technology College of Engineering, Kondwa, Pune, Maharashtra, India⁴

ABSTRACT: The attendance system was developed at schools, colleges, and universities to maintain discipline and guarantee that students could learn everything they could. There are two accepted ways to monitor students' attendance in a particular class. The roll number is called in the first, and pupils sign a sheet of paper next to their roll number in the second. In order to make this system more user-friendly, effective, and time-saving, it must be developed [1]. Using this automated technology, the teacher may take the class's attendance without being interrupted or wasting time. One of the many applications for the concept that will save time is facial identification. The main objective of this project is to build a robotic attendance system using a Raspberry Pi 3B+, Python libraries for OpenCV, and a recognizer algorithm. The proposed method can be used in any industry where an attendance system is required and existent. Furthermore, since the project's objectives and design requirements were all met, it is best to assert that this project is an engineering solution for all schools and institutions to track and manage attendance.[2] [3]

KEYWORDS:Text detection, Inpainting, Morphological operations, Connected component labelling.

I. INTRODUCTION

Every class in colleges has an attendance requirement. Frequently, classes that students enroll in have a minimal attendance requirement. The simplest methods for recording attendance include manually signing a piece of paper or using roll call. These operations take a lot of time, are exhausting, and don't involve any technology. Data from the attendance sheet must be manually entered into the database. In addition, it is easy to give someone else attendance by forging their signature or shouting out their name when roll is called. A popular and modern technique for streamlining attendance is biometric attendance. The biometric gadget can be connected to the database and update attendance automatically. to the uniqueness of thumb prints. The fact that children must wait in queue for biometric verification shows that this method stillwastes time. By passing the biometric device around the class, this issue can be overcome, although it can be distressing. All of the aforementioned attendance-taking techniques have the drawback that it is impossible to ensure that students will remain seated in class for the entire permitted period of time. A student may enter the classroom just before attendance or may leave it immediately after it.[1]

However, all of these problems—including the issue with roxy attendance that is common in institutions—are solved by facial recognition technology. Due to a camera's automatic presence detection, students no longer need to manually record their attendance. Since the camera is always taking pictures, it is easy to keep track of how much time a student spends in class. Final attendance will only be given if the student spent enough time in class, which might be determined by the professor teaching the class. Additionally, the system features a user-friendly interface where instructors can, if necessary, change the default attendance policies for a particular class or student and where students can check their attendance for any class or course.



II. RELATED WORK

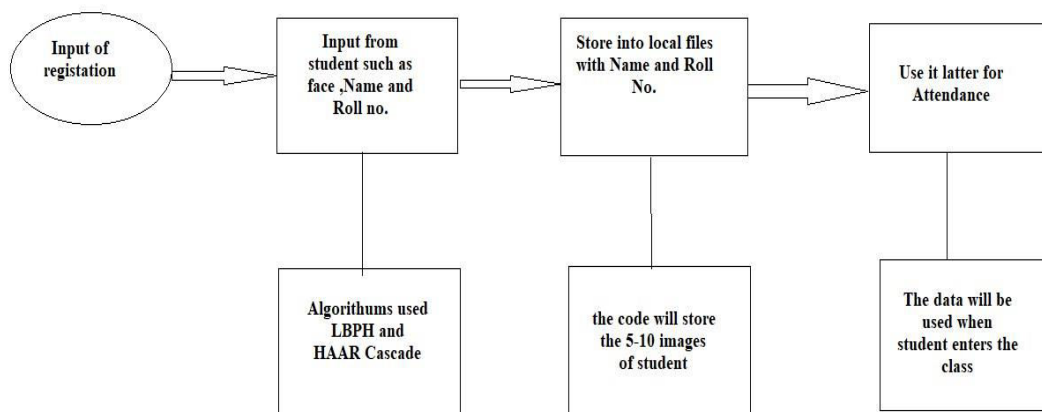
Numerous studies have been done so far on the various approaches that can be used to put in place an efficient attendance tracking system. The sorts of input techniques, types of data processing, and controllers utilized to build the systems all differ according to these approaches. Looking for the numerous solutions with the benefits and drawbacks of each system in this part. "NFC-based attendance system with a built-in camera for mobile devices" is the first system. A sort of short-range wireless communication called near field communication occurs when two devices—one active and the other passive are in close proximity to one another. The two devices are essentially electromagnetic induction-responsive inductor coils. In order to create an electromagnetic field with a specific radius and strength, the active device is used. [4] Which used to implement a system of attendance. For instance, in a classroom setting, pupils may be issued NFC tags that are individually programmed with their personal identification numbers. When a student first arrives for class, the professors bring NFC readers, and they ask them to tap their NFC tags on the reader—which might be the lecturers' phones by swiping them against it. The student's attendance is subsequently recorded in the school database using this information. [3] Nevertheless, this approach is susceptible to impersonation, where one individual could sign in as another.

Other related biometric systems that are used in universities, institutes, and schools are time management systems (Fingerprint recognition, RFID, etc.) to identify end users. These systems, however, raise new privacy issues. Additionally, users of these systems have the potential to physically harm them. As a result, they require more maintenance expenses. Our suggestion denies anyone physical access to the automated system. [1]

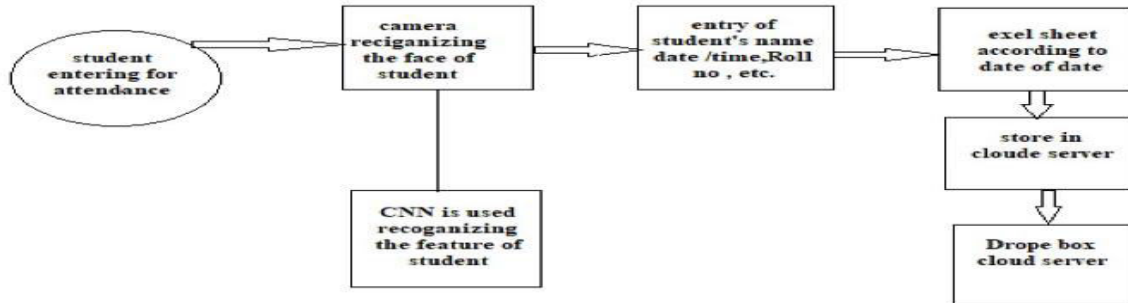
III. METHODOLOGY

We have thoroughly examined a variety of areas that are relevant to our project, thus we will base our prospective answer to our problem on the literature review. We'll provide a method in this section that will give you a broad overview of how to approach and carry out our project. We created this project in the most practical and effective way we could because the earlier work was insufficient. The recommended face detection module for this project uses OpenCV and Haar cascades. The facial recognition modules for this project are also advised to employ the CNN and KNN algorithms. [4],[5],[6]

Fig 1. Block diagram of Student registration (a), Block diagram of attendance entry (b)

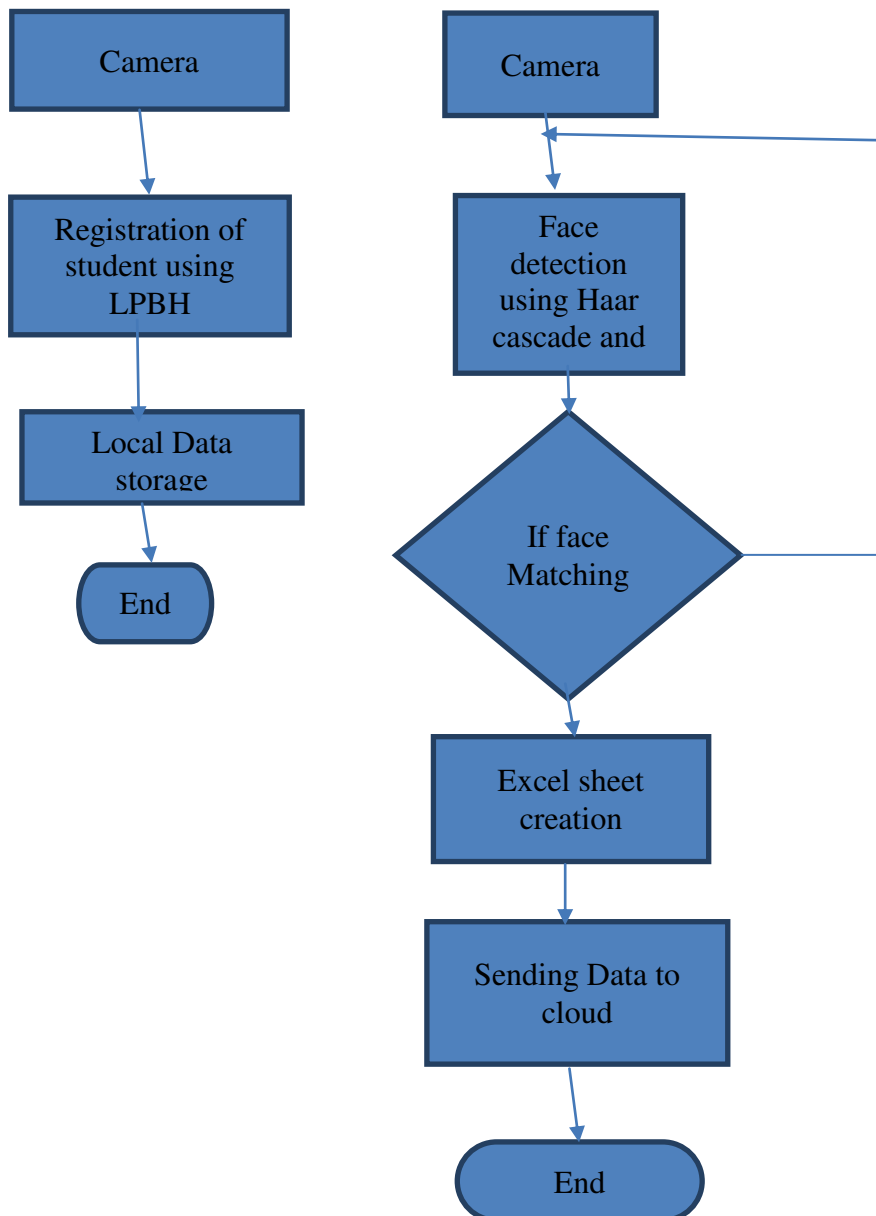


1. STUDENT REGISTRATION BLOCK DAIGRAM



2. ATTENDANCE ENTRY BLOCK DAIGRAM

Fig 2. Flow chart of system



Hardware Requirement

The required components to implement this project are as follows:

1. Raspberry Pi 3B+
2. 16 GB Micro SD Card SanDisk
3. LED touch screen
4. Arducam 8MP IMX219 Camera
5. Power supply cable
6. Mouse and keyboard
7. HDMI cable

Software Requirement

- **OpenCV (Python Library):**

OpenCV, which can be read and used on a range of platforms, including iOS, Android, and Windows, uses a number of programming languages, including Java, Python, and C++. Linux, Windows, and OS X. Additionally, based on CUDA and OpenCL, dynamically tailored interfaces for quick GPU tasks are being developed. To solve computer vision issues, the OpenCV-Python Python module was made. Current OpenCV

- **Code Software:**

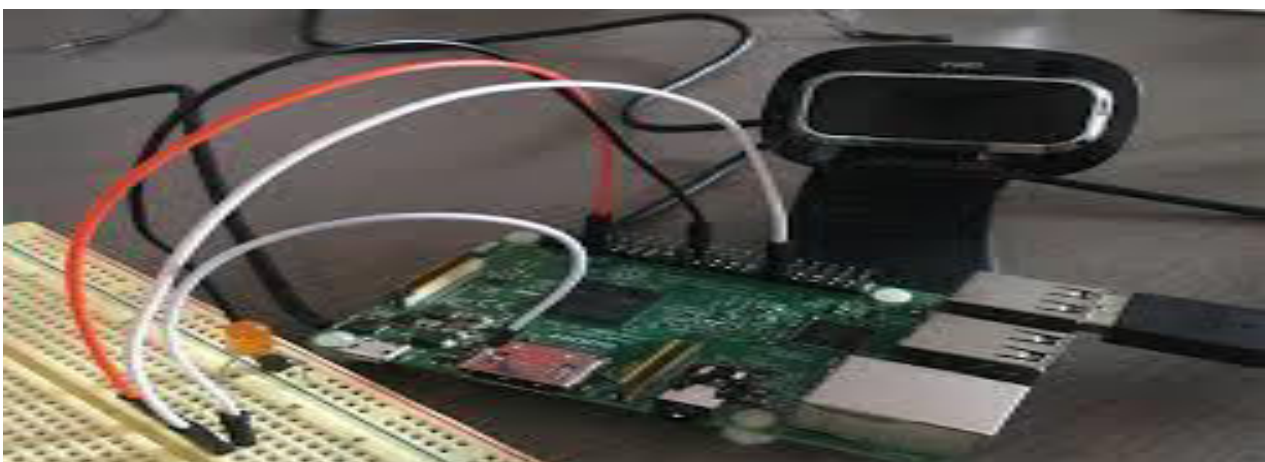
Windows A code manager known as visual studio code was produced by Microsoft and Linux. In essence, this technique aids in troubleshooting for Windows, integrates Language structure, intelligent code finishing, scraps, and code rewriting are all elements of GitHub and Git management. This I used in my project to execute Python code.

- **Tkinter (Python Library):**

The name of the GUI library that comes with Python is Tkinter. The construction of GUI apps is quick and simple thanks to the Python and Tkinter combo. It provides a powerful object-oriented interface for the Tk GUI toolkit. The de facto technique for creating Graphical User Interfaces (GUIs) in Python, it is included in all popular Python distributions. In fact, it's the sole framework present in the Python standard library.

Implementation

Fig 3. Hardware picture



Steps -

- 1) Installing the Raspberry PI in step one
- 2) Launch the attendance tracking system's GUI



- 3) Once the lecturer chose the correct day and schedule ID for the current class session, the attendance taking session may begin.
- 4) Launch a Python script in step three. A shell file is used to launch a Python script after the professor clicks a button in the main window.
- 5) The Python script with the training data is step four. When creating the face database, the training data (.yml) file is initially loaded and saved.
- 6) Acquire Faces At this point, the system will begin to take student portraits, which will then go through the same face detection and pre-processing steps.
- 7) Valid portraits that successfully identify the collected faces will then be compared to the recognizer's loaded gallery.
- 8) Register your attendance. A record of the most recent attendance will be entered to the attendance table kept by a database when the relevant student has been successfully identified through the capturing technique.

Fig 4. Code Snapshots

```
import numpy as np
import pickle
import cv2
import time

face_cascade = cv2.CascadeClassifier('cascades/data/haarcascade_frontalface_all2.xml')
eye_cascade = cv2.CascadeClassifier('cascades/data/haarcascade_eye.xml')
nose_cascade = cv2.CascadeClassifier('cascades/data/haarcascade_nose.xml')

tag_counter = 0
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainingdata/face-train.yml')

labels = {'personname': 1}
with open('pickles/face-labels.pickle', 'rb') as f:
    og_labels = pickle.load(f)
    labels = {v:k for k,v in og_labels.items()}

cap = cv2.VideoCapture(0)

while(True):
    # Capture frame-by-frame
    ret, frame = cap.read()

    # Convert to grayscale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.5, minNeighbors=5)

    for (x, y, w, h) in faces:
        # Crop the image
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = frame[y:y+h, x:x+w]

        # Detect faces
        id, conf = recognizer.predict(roi_gray)
        if conf >= 90:
            print(labels[id])

def loadDatabase():
    with open('database.csv', 'r') as f:
        mydataList = f.readlines()
        nameList = []
        for line in mydataList:
            entry = line.split(',')
            nameList.append(entry[0])

    if name not in nameList:
        row = dataLine.getRow()
        dtString = now.strftime("%d-%m-%Y")
        FaceIdLines["%s"%name], dtString
        statement = "insert into office" + name
        engine.say(statement)
        engine.runAndWait()

def CAPTURE_SCREEN_BETTER_THAN_WEBCAM
# def captureScreen(x=0, y=0, w=640, h=480):
#     cap = cv2.VideoCapture(0)
#     cap = cv2.VideoCapture(0)
#     return cap

encodetFrame = findFaceInImages()
print("Encoding complete")

cap = cv2.VideoCapture(0)

while True:
    success, img = cap.read()
    # tag = captureScreen()
    imgs = cv2.resize(img, (0, 0), None, 0.25, 0.25)
    imgs = cv2.cvtColor(imgs, cv2.COLOR_BGR2RGB)

    facesurFrame = face_recognition.face_locations(imgs)
    encodetFrame = face_recognition.face_encodings(imgs, facesurFrame)

    for encodetFace, faceLoc in zip(encodetFrame, facesurFrame):
        # ... (rest of the code) ...

recognizer = cv2.face.LBPHFaceRecognizer_create()
exists = os.path.isfile('trainingdata/face-train.yml')
if exists:
    recognizer.read('trainingdata/face-train.yml')
else:
    msg = "data missing, message-Please click on save profile to reset detail"
    return
harcascadePath = 'cascades/haarcascade_frontalface_default.xml'
faceCascade = cv2.CascadeClassifier(harcascadePath)

cap = cv2.VideoCapture(0)

font = cv2.FONT_HERSHEY_SIMPLEX
col_names = ['id', 'name', 'date', 'time', 'status']
exists = os.path.isfile('studentdetails/studentdetails.csv')
if exists:
    df = pd.read_csv('studentdetails/studentdetails.csv')
else:
    msg = "details missing, message-Students details are missing, please check!"
    com.release()
    cv2.destroyAllWindows()
    window.destroy()

while True:
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.2, 5)

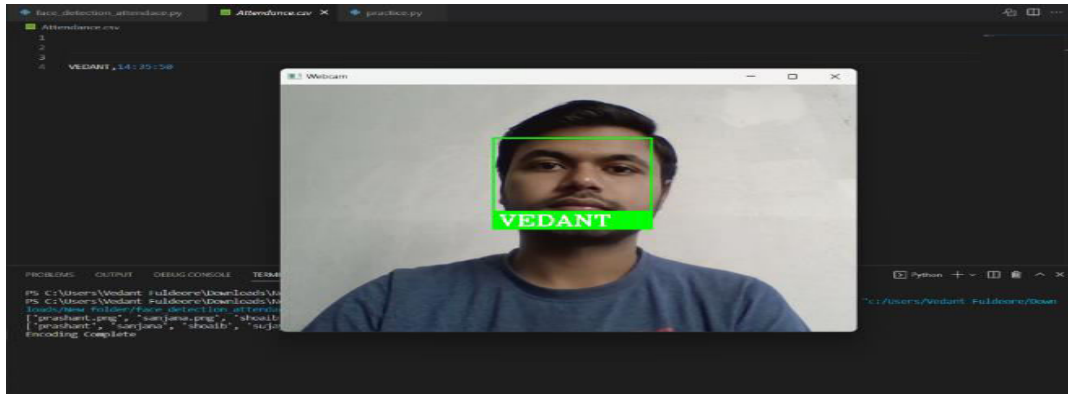
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)
        serial, conf = recognizer.predict(gray[y:y+h, x:x+w])
        if conf < 90:
            ts = time.time()
            date = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y')
            time_stamp = datetime.datetime.fromtimestamp(ts).strftime('%d-%m-%Y %H:%M:%S')
            as = df.loc[df['serial no.']== serial]['name'].values
            id = df.loc[df['serial no.']== serial]['id'].values
            id = str(id)
```

IV. EXPERIMENTAL RESULTS

The results of testing the face detection algorithm on a number of volunteers show that it can create a GUI interface with an accuracy of roughly 60%. The act of completing all the implementation steps are as follows. The implementation test results are listed below, and they indicate that one student was identified and registered as present during the attendance taking procedure and saved in an excel file which is then uploaded to cloud.

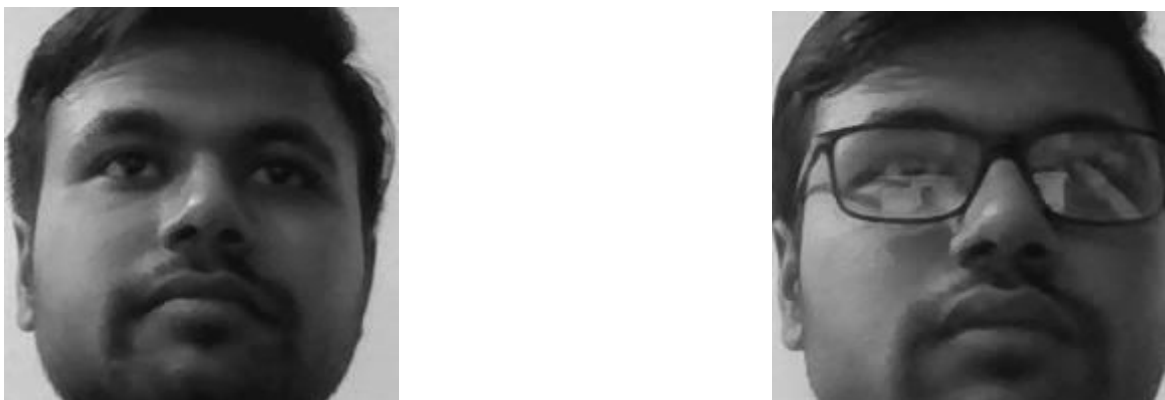


Fig 5. Result of the device registering the student



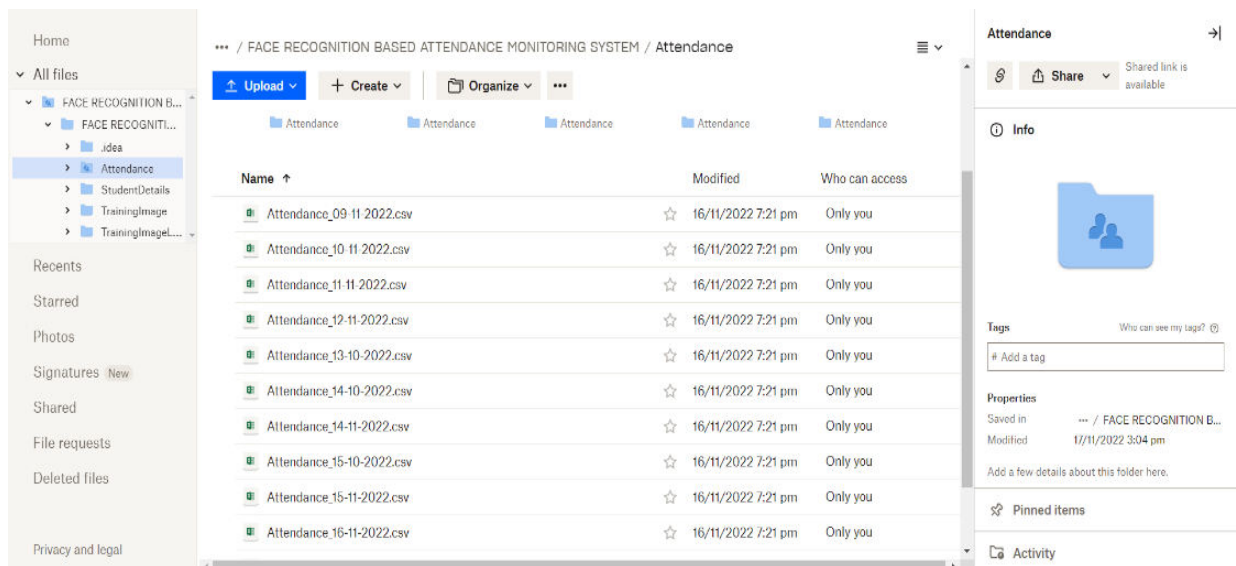
Also, we noticed some error. Like if the person is wearing glasses, his/her name is recognized sometimes but other time not, in Fig 6

Fig 6. Error Images, with glasses and without glasses



In our project, we used the cloud platform Dropbox to keep the students' attendance, as illustrated in Fig 7

Fig 7. Snapshot of attendance being stored in Cloud server dropbox





V. CONCLUSION

This study recommends a novel face recognition-based method for assessing and analyzing attendance in real-time. The attendance for each class is determined automatically, with no human input. The system makes sure that a pupil must be in frame for at least a certain amount of time in order to be marked present, and that improvement is always possible. The system may occasionally fail to recognize students or identify them incorrectly due to the limits of face recognition technology. The lighting in the classroom and the positioning of students' faces are examples of outside variables that may affect how accurate the face recognition system is. Recent research has resulted in enhanced facial recognition algorithms that are more robust and adaptable to varied contexts, which enhances the suggested system.

REFERENCES

- [1] Smitha; Pavithra S Hegde; Afshin "Face Recognition based Attendance Management System" 2020
- [2] Okokpujie, Kennedy O., et al. "Design and implementation of a student attendance system using iris biometric recognition", International Conference on Computational Science and Computational Intelligence (CSCI), IEEE, 2017.
- [3] Face Recognition Attendance System – Biometric Facial Recognition System. [online] Available at: <https://www.matrixaccesscontrol.com/facerecognition.html> [Accessed 15 Oct. 2018].
- [4] Rahul Chauhan; Kamal Kumar Ghanshala; R.C Joshi "Convolutional Neural Network (CNN) for Image Detection and Recognition" 2016
- [5] Teddy Mantoro; Media A. Ayu; Suhendi "Multi-Faces Recognition Process Using Haar Cascades and Eigenface Methods" 2018
- [6] Akshata Uday Naik; Nitesh Guinde "LBPH Algorithm for Frontal and Side Profile Face Recognition on GPU



Impact Factor: 8.423



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details