INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN

Computer Science & Artificial Intelligence Department

IT WORKSHOP

Attendance Tracking System

Using Face recognition

Submitted To:

Ms.Dimple
Assistant Professor
Department of CSAI

Submitted By:

Karishma Jain (06501172021)
Radhika Bansal (01601172021)
Tejaswi Tyagi (07401172021)

Attendance Tracking System

* Using face recognition *

…………………………………………………………………………………………………

Karishma Jain (06501172021)

Radhika Bansal (01601172021)

Tejaswi Tyagi (07401172021)

…………………………………………………………………………………………………

Abstract — *The main objective of this project is to build a facial recognition and identification system for tracking students' attendance to overcome the ambiguities caused by traditional attendance tracking systems. In this project, face databases will be created to pump data into the recognizer algorithm. Then, during the attendance-taking session, faces will be compared against the database to seek identity. When an individual is identified, their attendance will be taken down automatically, saving necessary information into a sheet.*

Keywords — machine-learning, face-recognition, face-landmarks, CNN, face-detection, OpenCV, Numpy, face-encodings, attendance

I. Introduction

I.I Background

The primary purpose of this project is to build a face recognition-based attendance monitoring system for educational institutions to enhance and upgrade the current attendance system into more efficient and effective than before. Consider a conventional attendance tracking system, how cumbersome it is to mark the attendance for everyone manually! This itself is a time-consuming task and moreover, the accuracy can also not be guaranteed as a third party may mark the attendance for the absentees.

For example, student A is too lazy to attend a particular class, so student B helped him/her to sign for the attendance which in fact student A didn't attend the class, but the system overlooked this matter due to no enforcement practice. Supposing the institution establishes enforcement, it might need to waste a lot of human resources and time which in turn will not be practical at all. Thus, all the recorded attendance in the previous system is not reliable for analysis usage.

I.II Motivation

The main intention of this project is to solve the issues encountered in the old attendance system while reproducing a brand new innovative intelligent system that can provide convenience to the firm. In this project, an application will be developed which is capable of recognizing the identity of each individual and eventually record down the data into a CSV file.

Students will be more punctual in attending classes. This is due to the attendance of a student can only be taken personally and any absentees will be noticed by the system. This can not only train the student to be punctual as well as avoids any immoral ethics such as signing the attendance for their friends. The institution can save a lot of resources as enforcement is now done by means of technology rather than human supervision which will waste a lot of human resources in an insignificant process. The application can operate on any device at any location as long as there is Wi-Fi coverage or Ethernet connection which makes the attendance system portable to be placed at any intended location. For example, the device can be placed at the entrance of the classroom to take attendance. It saves a lot of cost in the sense that it eliminates the paperwork completely. The system is also time effective because all calculations are automated. In short, the project is developed to solve the existing issues in the old attendance system.

II. Literature Survey

Paper	Objective	Dataset	Result	Conclusion/Future Scope
Face Recognition by Elastic Bunch Graph Matching.[1]	Recognizing human faces from single images out of a large database containing one image per person.	FERET database and the Bochum database.	One recognition against a gallery of several hundred models takes approximately 30 seconds on a SPARCstation 10-512.	Too many degrees of freedom which can be reduced could be reduced by using typical distortions. This can increase recognition performance.

Hidden Markov Models for Face Recognition [2]	Using Hidden Markov Models (HMMs) for face training and recognition.	Olivetti Research Ltd (ORL) database of faces.	Each human face is modeled using a one-dimensional Hidden Markov model (HMM).	Experimental results show that the feature extraction using DCT gives better recognition accuracy than the other two proposed methods (DWT, PCA).
Attendance System Using NFC Technology with Embedded Camera on Mobile Device [3]	Speeding Attendance tracking using NFC technology and mobile application	The student images are collected using a browser to the face training module (JAVA+ JSP)	On successful validation of the student, the application marks the attendance of the student in the application. This attendance is marked on the central server. Likewise, notification is also sent to the parents of the students for their reckoning.	For future developments of NFC- based applications it will guide us to believe that underlying each operating mode benefits by analyzing already developed applications.
Face Recognition Using Neural Network: A Review [4]	Discussing the face recognition methods, and algorithms proposed by many researchers using artificial neural networks (ANN)	Various research papers	This paper includes a summary review of literature studies related to face recognition systems based on ANNs.	In future work, a face recognition system will be based on real data with a hybrid Wavelet and ANN approach with many hidden layers. Different network architectures and parameters‟ values of BPNN will be used to determine the result in the best performance values of the face detection system.

Smart Attendance Management System Using Face Recognition [5]	The daily attendance of students is recorded subject wise which is stored by the administrator.	A dataset of students' images in different poses and variations along with 128-d facial features.	The faces which are not in our training dataset are marked as unknown. The attendance of recognized images of students is marked in real-time.	In the future, this system needs to be improved to recognize students from some distance with greater accuracy.
Face Recognition Based Attendance Management System With Raspberry Pi 2 Using Eigenfaces Algorithm	Develop an attendance management system that uses the face of students as the feed input.	—	Accurate attendance information of the students in easy way and upload the attendance into server using Ethernet cable.These systems develop outputs with 88 percent of accuracy.	When the number of students' faces increases the accuracy will decrease slightly. This can be worked upon.

III. Methodology

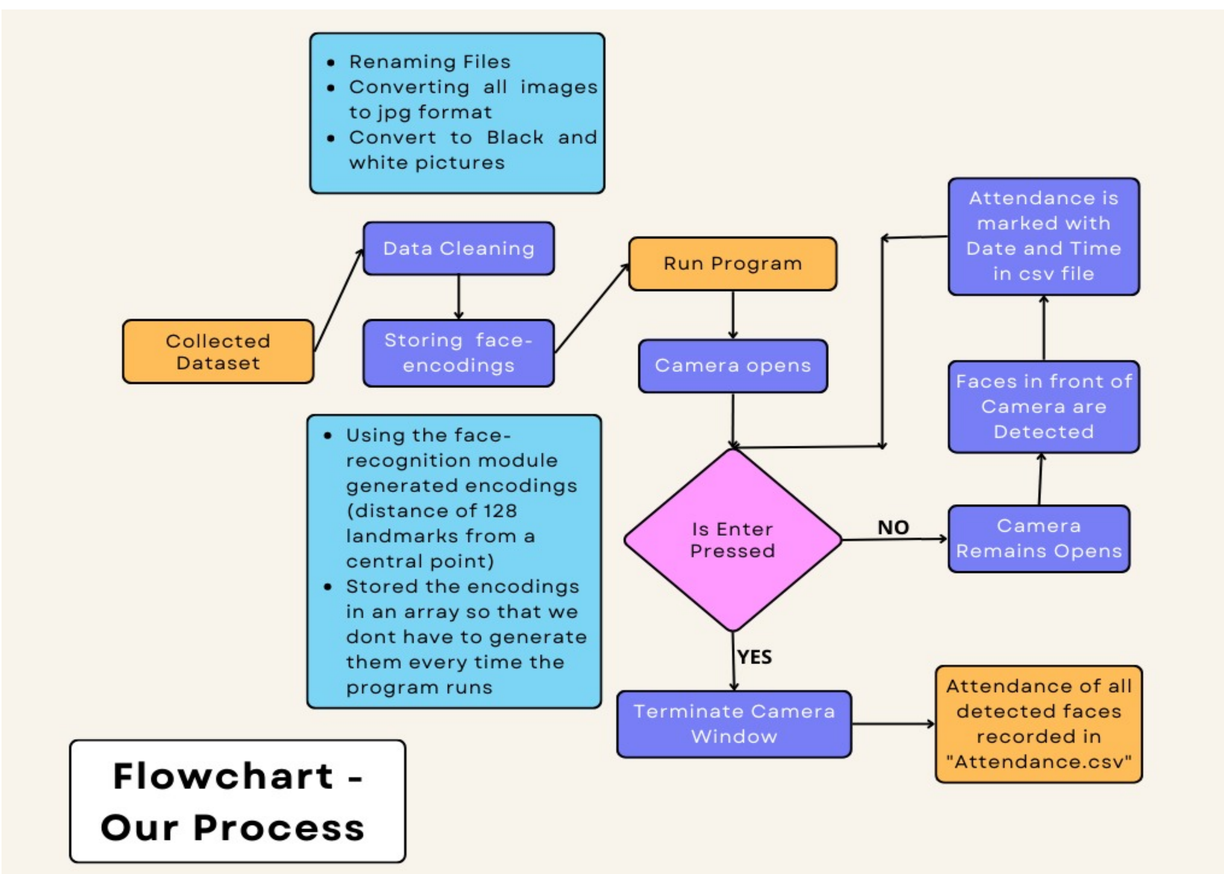
III.I Primary Working

This model uses facial recognition to detect and identify faces for tracking attendance. But facial recognition is not seen as a single-level problem for the computer. [6]

It involves a series of issues that are to be solved:

1) First, capture the picture or frame for the facial recognition model to work on and find faces in it.
2) Second, focusing on each face detected and developing the ability to understand that even if a face is turned in a weird direction or in bad lighting it is still the same person.
3) Third, to be able to pick out unique features of the face that you can use to tell it apart from other people— like how big the eyes are, how long the face is, etc. [7]
4) The fourth and final stage is comparing these detected faces with the faces already stored in the database.

The above-mentioned four points give us a rough idea about how this system works. The more technical discussion on the working of this model is highlighted in upcoming sections.



The complete process of this project

III.II Architecture

Step 1: Finding all the Faces

The first step in our pipeline is *face detection*.[8] Obviously, we need to locate the faces in a photograph before we can try to tell them apart.

To find faces in an image, we'll start by making our image black and white because we don't need color data to find faces: Then we'll look at every single pixel in our image one at a time. For every single pixel, we want to look at the pixels that directly surround it: Our goal is to figure out how dark the current pixel is compared to the pixels directly surrounding it. Then we want to draw an arrow showing in which direction the image is getting darker: If you repeat that process for every single pixel in the image [9], you end up with every pixel being replaced by an arrow. These arrows are called *gradients* and they show the flow from light to dark across the entire image:

If we analyze pixels directly, really dark images and really light images of the same person will have totally different pixel values. But by only considering the *direction* that brightness changes, both really dark images, and really bright images will end up with the same exact representation. That makes the problem a lot easier to solve [10]

Step 2: Posing and Projecting Faces

When we isolated the faces in our image. But now we have to deal with the problem that faces turned in different directions look totally different to a computer: To do this, we are going to use an algorithm called face landmark estimation. The basic idea is we will come up with 68 specific points (called *landmarks*) Now that we know where the eyes and mouth area are, we'll simply rotate, scale, and shear the image so that the eyes and mouth are centered as best as possible.[11] Now no matter how the face is turned, we are able to center the eyes and mouth in roughly the same position in the image. This will make our next step a lot more accurate.

Step 3: Encoding Faces

The solution is to train a Deep Convolutional Neural Network. But instead of training the network to recognize pictures of objects like we did last time, we are going to train it to generate 128 measurements for each face. Machine learning people call the 128 measurements of each to face an embedding.

Step 4: Finding the person's name from the encoding

This last step is actually the easiest step in the whole process. All we have to do is find the person in our database of known people who has the closest measurements to our test image.

Flowchart depicting the steps in our recognition process

III.III Libraries Used

OpenCV [12]
It is a huge open-source library for computer vision, machine learning, and image processing. It can process images and videos to identify faces.

- CV2.imread
- CV2.cvtColor
- CV2.VideoCapture
- CV2.resize
- CV2.rectangle
- CV2.putText
- CV2.imshow
- CV2.destroyAllWindows
- CV2.waitkey

Numpy
We just need it to convert our images into some form of an array so that we can store the model that has been trained. It provides a multidimensional array object, as well as variations such as masks and matrices, which can be used for various math operations.

- numpy.argmin

Face_recognition - The face recognition library has many methods (functions) to deal with faces in images and one of them is known as face_locations which will find the face's locations inside a particular image.

- face_recogniton.face_encodings
- face_recognition.face_locations
- face_recognition.compare_faces
- face_recognition.face_distance

Os - Used for creating and removing a directory (folder), fetching its contents, and changing and identifying the current directory.

- os.listdir
- os.path.splitext

Datetime - Keeping the track of current date and time to update the real-time data in the spreadsheet to track attendance.

- strftime
- datetime.now()

III.IV Landmarks And Face Alignment

This image was created by Brandon Amos of CMU who works on OpenFace

The basic idea is we will come up with 68 specific points (called *landmarks*) that exist on every face — the top of the chin, the outside edge of each eye, the inner edge of each eyebrow, etc. Then we will train a machine-learning algorithm to be able to find these 68 specific points on any face.

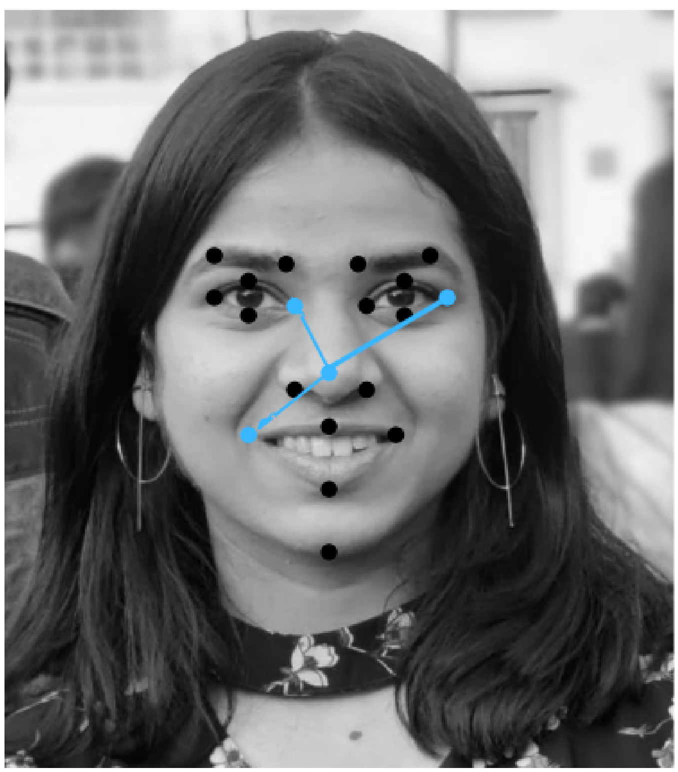
Now that we know where the eyes and mouth are, we'll simply rotate, scale and shear the image so that the eyes and mouth are centered as best as possible. We are only going to use basic image transformations like rotation and scale that preserve parallel lines. Now no matter how the face is turned, we are able to center the eyes and mouth in roughly the same position in the image.

Relating The Landmarks

The most accurate approach is to let the computer figure out the measurements to collect itself. The solution is to train a Deep Convolutional Neural Network.[13] We are going to train it to generate 128 measurements for each face. The training process has three steps: upload a training picture, upload another pic and then upload a picture of a totally different person. Then the algorithm looks at the measurements it is currently generating for each of those three images. It then tweaks the neural network slightly so that it makes sure the measurements it generates for #1 and #2 are slightly closer while making sure the measurements for #2 and #3 are slightly further apart.

III.V Encoding Faces

Unlike the human brain, the computer uses smart techniques to work on facial recognition and improve its results or efficiency. Here comes the concept of the Euclidean distance can be used to calculate the distance between any two points in two- dimensional space, and also to measure the absolute distance between points in N-dimensional space. For face recognition, smaller values indicate more similar faces. [8] The feature points representing facial features were calculated, and the image to be detected is processed by a function to obtain a 128-dimensional face feature vector, this constitutes the condition for face similarity calculation under Euclidean distance. To achieve this, the '.face_distance' function of the face recognition library is used.



This image was created to show how encodings are calculated between landmarks

IV. Limitations of Model 1

- Speed: This model is slow to run. And takes time to catch frames and images and then the processing times add to it.

- Google Colab: While running this model on Google Colab we are required to install the libraries and dlib module, again and again, every time we want to execute the code.

- Dataset Loading: Our primary model was run on a smaller dataset and the results generated were captured. Though the dataset loaded was concise the model still took more time than expected.

- No multi faces capturing: A major drawback of the present model is that it was not able to capture and mark the attendance if multiple faces were encountered in a single frame.

- Repeated generation of encodings: As soon as the face is detected by the model landmarks and encodings for each face present in the dataset i.e 128 encodings are generated every time. Ultimately, consuming a lot of time in the process.

- Infinite attendance marked: As long as the camera is in an active state, it keeps detecting the same person again and again and marks the attendance multiple times.

V. Refining This Model

Our previous model 1 had various limitations that forced us to further work on building a model that overcomes them and is more efficient than the previous model. Though the skeleton of Model 2 is similar to that of Model 1, various refinements are introduced. Model 1 was extremely slow in working on a dataset as small as containing 2 images, along with that it was marking attendance multiple times if the face is encountered again on a particular day. Moreover, while running this model on Google Collab we are required to install the libraries and dlib module, again and again, every time we want to execute the code. All these resulted in the reduction of efficiency of the ML model.

V.I Datasets

1 Initial Dataset - Model 1

Our initial dataset had the 3 images of the members of our project in .jpg format. Since in model 1 the encodings for each of the images in the dataset are generated every time we run the model for a face, the process took a lot of time.

2 Dropping Kaggle's Dataset

We initially thought of implementing the model through the Kaggle dataset but the idea was later dropped. The reasons behind not using the Kaggle Dataset are:

- The dataset would require an extensive amount of pre-processing to meet the requirements of our model i.e conversion of images into .jpg format, changing the nomenclature for each image in the dataset, etc.

- Even if we are able to successfully achieve the pre-processing milestone, we will not be able to check whether the model is working well or not, because the images stored are of no real persons we can test the model on.

3 Building Our Own Dataset

So to bring accuracy and efficiency to the dataset we finally decided to build the dataset of the images of the students from the CSAI batch itself. This dataset, therefore, gives a more real-time feel to the project, and also the data pre-processing is further reduced. We are then doing data cleaning on the dataset we collected. This includes-

1) Renaming the file with proper names such that it aligns with the given format of the model.
2) converting the file format to jpg for the face recognition model to work.
3) converting the images to black and white images through the inbuilt features of the face-recognition module so that the images can be distinguished based upon the intensity of pixels to generate gradients.

Link to the Dataset:
https://drive.google.com/drive/folders/13f5fXvym8ad-ST8vHkyY-t-gKtBd7Yc5?usp=sharing

V.II Result Analysis For Model 2

Storing The Encodings Now

Unlike our previous model that generated the 128 encodings and landmarks each time a face was encountered in the camera, our present model now stores these encodings to overcome the time of implementation. Now the model is able to efficiently recognize the people from the dataset and that too takes less time now.

Muitiple Faces Are Detected In Single Frame

Further boosting the efficiency of the model, our system is now able to detect multiple persons in a single frame and mark their attendance simultaneously in the attendance CSV file.

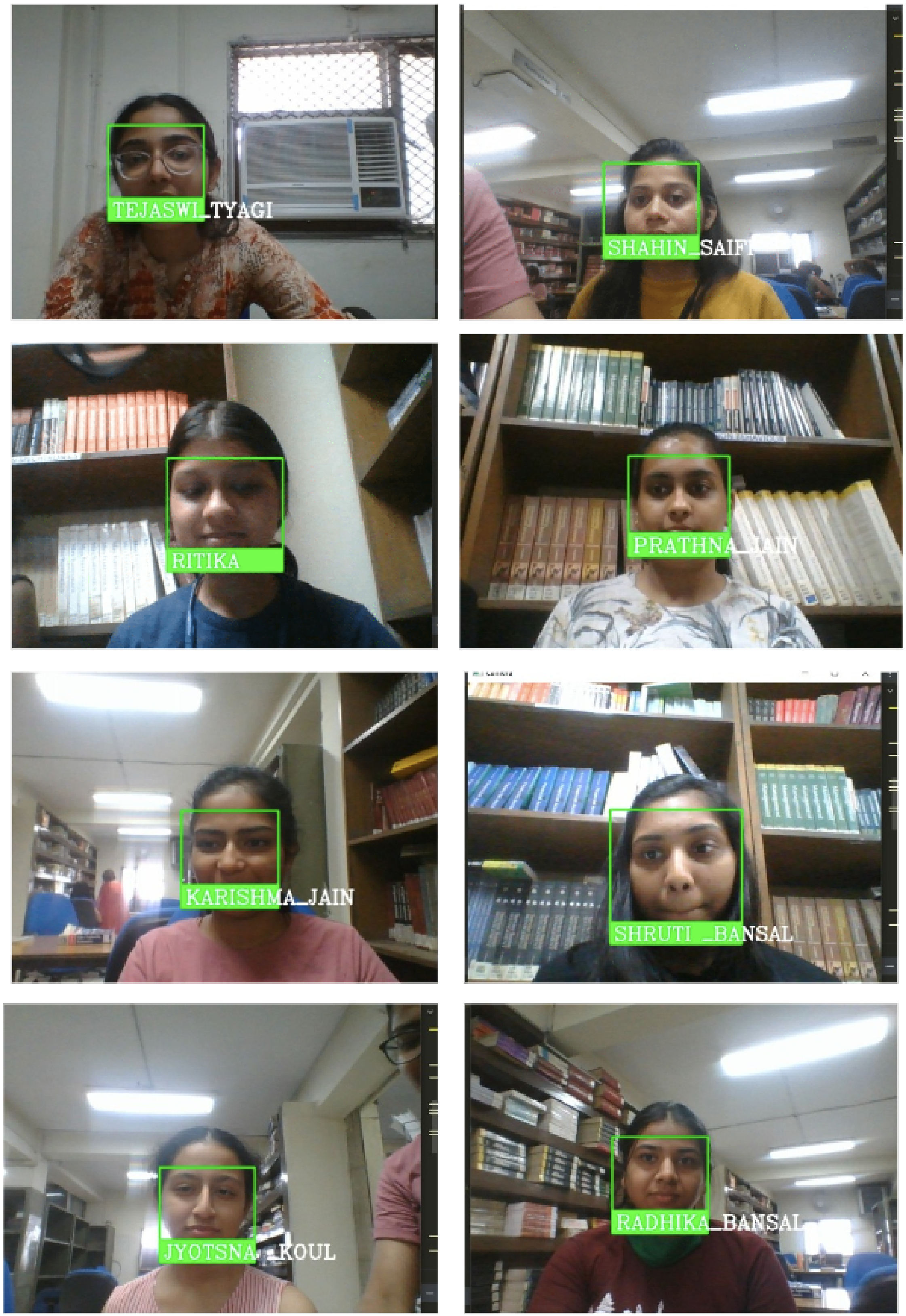
No Multiple Attendance Marked

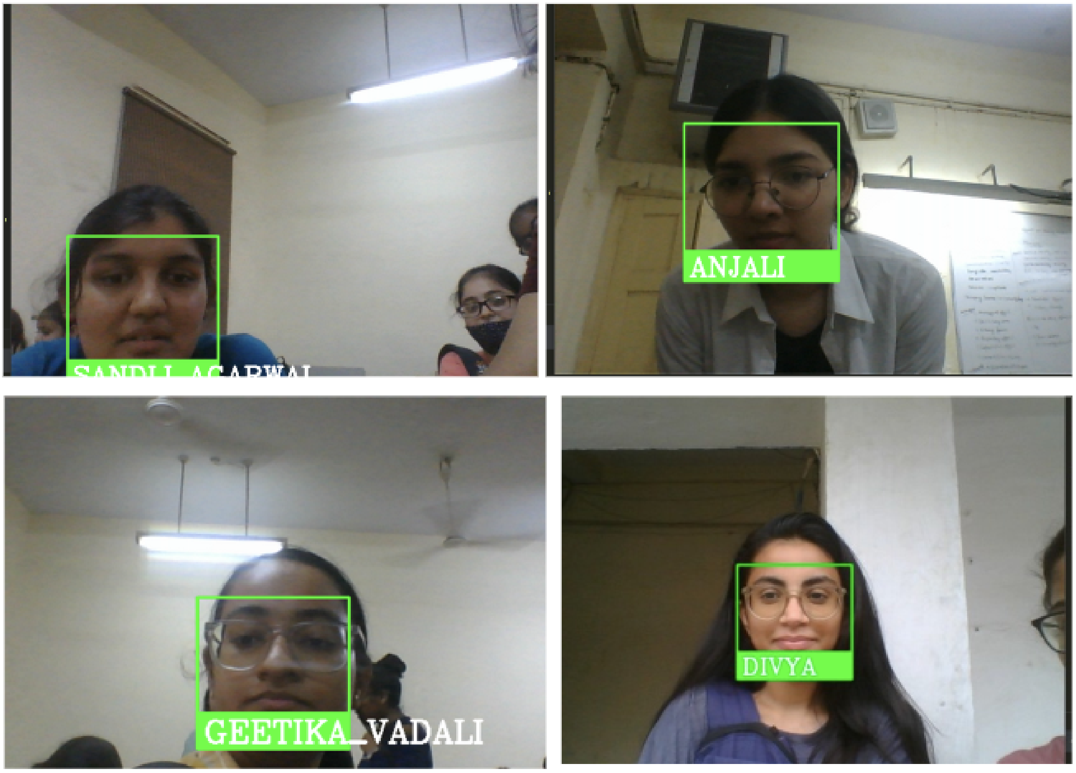
The new model now also overcomes the problem of marking multiple attendances of a person. So if a person's attendance is already marked in the file, it will not mark the attendance again for a particular day.

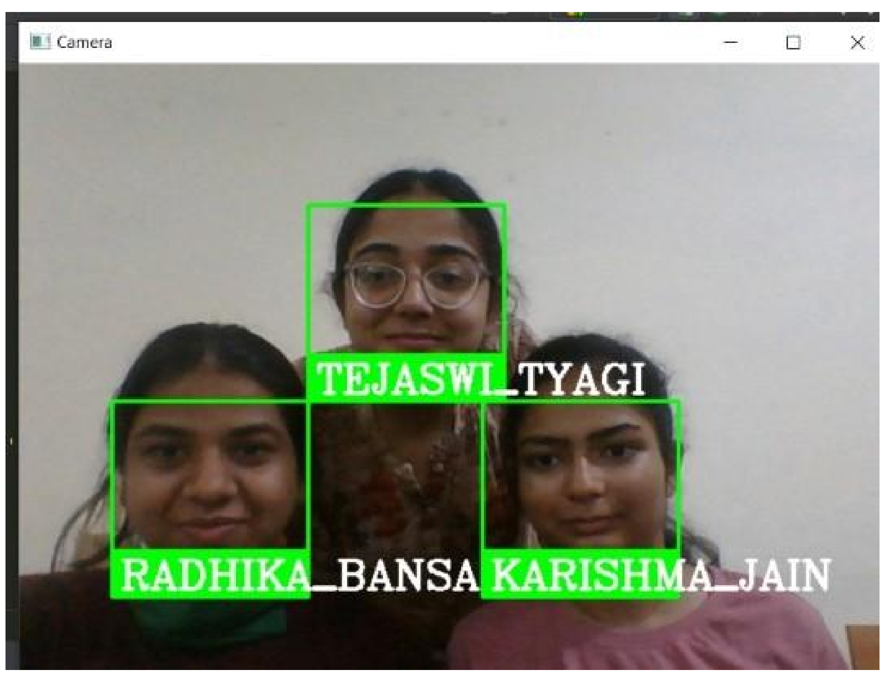
VI. Result

Through this project of the Attendance tracking system using face recognition, we finally are able to generate a CSV file containing the records of students with their Names, Date, and Time. This sheet can now be used to maintain the attendance records of the students of our class. Further, this sheet can be emailed or sent as an SMS to share the details with others.

Attached below are the images showing our real-time working model on our dataset i.e. the students of our class CSAI 1 and 2.


TEJASWI_TYAGI
SHAHIN_SAIFI
RITIKA
PRATHNA_JAIN
KARISHMA_JAIN
SHRUTI _BANSAL
JYOTSNA _KOUL
RADHIKA_BANSAL


ANJALI
GEETIKA_VADALI
DIVYA


Camera
TEJASWI_TYAGI
RADHIKA_BANSA
KARISHMA_JAIN

VII. Conclusion And Future Scope

Increasing User Experience

To pump up the user experience and accuracy of the model, we can bring further enhancements so that even if the face detected is in extreme motion, the attendance is still accurately marked.

Sending The Attendance File To The Admin

We wish to further add an automatic email sending feature that will send a copy of the attendance file to the respective owner automatically. As of now, the CSV file of the attendance created can be manually mailed to the owner.

VIII. References

[1] Laurenz Wiskott , Jean-Marc Fellous , Norbert Kr¨uger, "Face Recognition by Elastic Bunch Graph Matching", Institute for Neural Computation Ruhr-University Bochum Germany ,2000

[2] F. H. Alhadi, W. Fakhr and A. Farag,"Hidden Markov Models for Face Recognition",Computer Engineering Institute of Technology,Atlanta

[3] Bhise, Khichi, Korde,Lokare "Attendance System Using NFC Technology with Embedded Camera on Mobile Device",International Journal of Advanced Research in Computer and Communication Engineering, February 2015

[4] Manisha M. Kasar , Debnath Bhattacharyya and Tai-hoon Kim, "Face Recognition Using Neural Network: A Review " International journal of Security and its Application .March 2016.

[5] Kaneez Laila Bhatti , Laraib Mughal, "Smart Attendance Management System Using Face Recognition ",2018

[6] Francis Galton, "Personal identification and description," In Nature, pp. 173-177, June 21, 1888.

[7] W. Zaho, “Robust image based 3D face recognition,” Ph.D. Thesis, Maryland University, 1999.

[8] R. Chellappa, C.L. Wilson and C. Sirohey, ”Human and machine recognition of faces: A survey,” Proc. IEEE, vol. 83, no. 5, pp. 705- 740, may 1995.

[9] T. Fromherz, P. Stucki, M. Bichsel, “A survey of face recognition,” MML Technical Report, No 97.01, Dept. of Computer Science, University of Zurich, Zurich, 1997.

[10] M. Turk and A. Pentland “Face recognition using euclidean distances,” In Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–591, 1991.

[11] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, “Learning rich features from RGB-D images for object detection and segmentation,” Proc. Eur. Conf. Comput.Vis., pp. 345–360, 2014.

[12] OpenCvDocumentation -https://opencv.org

[13] K. Simonyan and A. Zisserman, “Very deep convolutional networks