FACE RECOGNITION WITH REALTIME DATABASE

T *Karthikeya*
Department of AIT-CSE
Chandigarh University
Kharar, Punjab, India
21BCS6274@cuchd.in

L Hari Chandraprasad Naik
Department of AIT-CSE
Chandigarh University
Kharar, Punjab, India.
21BCS6116@cuchd.in

Vignan Kumar
Department of AIT-CSE
Chandigarh University
Kharar, Punjab, India
21BCS6323@cuchd.in

Shiva Reddy
Department of AIT-CSE
Chandigarh University
Kharar, Punjab, India
21BCS6003@cuchd.in

Dr Priyanka Kaushik
Department of AIT-CSE
Chandigarh University
Kharar, Punjab, India
Priyanka.E13618@cuchd.in

Abstract— **This study's goal is to develop a face recognition system based on real-time video. This article basically establishes four guidelines for considering the issue: transmitting the cost of video time in percent to confirm participation, security of video time of real video, accuracy of the system at login, and a real-time video processing system. The effectiveness of AFR technology has increased significantly over the past few years and these systems are now used for security and commercial purposes. Student participation is an important task in the classroom. When done manually, it usually takes a lot of lesson time that can be used for learning. An auto-enrollment system using facial recognition is a suggested solution to an existing problem. The face is an important part of a person. This tutorial shows how to detect and recognize human faces in real time using Arduino UNO. This project uses the open-source graphics library OpenCV to define an efficient algorithm. Face Recognition, FaceProgression, Face Training, Face Recognition and Interaction Data are the five models that make up our approach. We built a face database to recognize students' faces. The student database contains the faces of all students and is initially used to train the algorithm. The system uses a user-friendly interface to maximize the user experience while collecting student images and engagement throughout training and testing. The program can also use many applications that can use facial recognition for authentication. Using Arduino UNO can reduce product cost andincrease performance because it can be connected to any device for now. The system automatically sends messages to the department heads and advisors of the students who are absent, informing them about the student attendance status and updating their student records. The unique thing about our model is that it has poll signing capability; this means that the student must be in the classroom for at least 20 or 30 minutes out of a total of 50 minutes before our signature model for students. Participating in facial recognition can reduce meeting times by up to 60%.**

Keywords— Automatic face recognition, attendance system, Arduino UNO, OpenCV, face detection, face preprocessing, face ***training, face recognition, attendance database, user interface, authentication, cost-effective, time quantum.***

Introduction

1) A. Problem Definition

Due to the growth of the Internet, computer technology has penetrated many aspects of people's lives and jobs. Today, people are increasingly interacting with computers. Also, more and more people use computers regularly. Withal strong creative spirit, one of the most challenging projects in the field offers a wide range of potential applications. With time, facial recognition technology has become a vital identification mark that makes it easier to tell one person from another. Facial recognition is the result of combining computers and artificial intelligence. Its highly sophisticated innovations and wide applicability make it one of the most complex subjects in its field. The current attendance system is manual. This loses a lot of time for both teachers and students. Students may have to wait longer if attendance is recorded manually. Even if attendance is manually recorded, trusted individuals can still attend classes.Human error is an ongoing cost of manual visits. Everyone isrecognizable only by their faces. Therefore, automating the attendance process increases class productivity. I chose Arduino UNO 3 for face recognition so that it can be used on all platforms. The webcam connects to the Arduino UNO module. Face identification differentiates between recognized and unrecognized faces. This module can be used for a variety of purposes, such as face recognition to verify. With the suggested approach, we collect attendanceby employing face recognition, which can identify each student's face while they are in class. The study revealed that the existing fingerprint attendance system, which it used as an example, has an error rate of roughly 5% and that there would be a phenomenon where fingerprints cannot be struck, which has a significant impact on attendance efficiency, particularly in big attendance locations. Face recognition technology is being used in the classroom to assist reduce the number of mischievous kids who use technology as a means of proxy, such as scanning their fingers to skip class.

2) B. Problem Overview

The manual attendance technique used in schools today is ineffective and can result in mistakes and wasted time for both teachers and pupils. It is suggested that the usage of facial recognition software be used as a method to automate the attendance process, which can boost class efficiency and decrease disruptive behavior. The current fingerprint attendance method can be influenced by things like the inability to hit fingerprints, which can lower attendance efficiency at large sites. It has a 5% error rate. The suggested method may be able to get over these restrictions and offer a wide range of possible uses beyond attendance by utilizing facial recognition technology based on the Arduino UNO 3 module and a webcam. Before being used, facial recognition technology must first be properly reviewed from an implementation and ethical standpoint.

II. LITERATURE REVIEW

Here is a literature review of some recent research papers on facial recognition using real-time data for student participants:

[1]. A. Kumar et al., A Time-Based Approach to Student Care. (2021) - This article presents real-time facial recognition to monitor student engagement using deep learning techniques and real-time data. The system was tested on a database of students' photos and accuracy was achieved.

[2]. " System that Real-Time Recognizes Faces to Track Students' Attendance Using Raspberry Pi", P. Jain and P. Patil (2020) - This document describes real-time facial recognition for student engagement using Raspberry Pi and real-time data. . The system was tested on a database of students' photos and accuracy was achieved.

[3]. "Real-time Face Recognition System for Student Attendance Tracking Using IoT" by S. Parveen and S. S. Hussain (2020) - This article presents a real-time facial recognition system for student attendance tracking using IoT technology and real-time tracking. The system was tested on a database of students' photos and accuracy was achieved. [4]. "Real-Time Student Attendance Monitoring System Using Face Recognition and Machine Learning" by V. Singhand R. Singh (2021) - This article presents real-time student attendance tracking using facial recognition information and machine learning. The system was tested on a database of students' photos and accuracy was achieved.

Together, these data show that real-time facial recognition using deep learning techniques, IoT technology and edge computing can monitor student engagement at home. Using a real-time database allows the system to process images quickly and accurately.

B. Proposed System

Real-time face detection involves real-time face recognition and facial detection in video streams. The basic steps involved in real-time face recognition are:

1. Face recognition. The first step is recognizing faces in the video stream. This can be done using deep learning-based face detectors such as MTCNN, which can detect faces even under complex lighting conditions and from various angles, or using techniques such as Haar Cascade.
2. Face Alignment: When a face is detected, it may need to align to a standard pose for the recognition algorithm to work effectively. This includes positioning, sizing and orienting the face so that it is centered and facing forward.
3. Feature extraction. When the faces are aligned, elements are extracted from the faces. This is usually done through deep learning-based techniques such as face embedding, which encodes unique facial features into compact vector representations.
4. Face recognition. The embedded faces are then compared to a database of known faces to identify people in the video stream. This is usually done through methods such as nearest neighbor search or clustering, which can efficiently find the closest matching face in a database.
5. Database update: If a video stream encounters a new face that is not yet in the database, the user may be prompted to add the new face to the database for later recognition.
6. Real-time feedback. Finally, the results of the Real-time face recognition system display include overlaying names or information about detected faces on a video stream.

The challenging task of real-time face recognition necessitates the use of quick and precise algorithms. Careful selection and optimization of each pipeline stage is critical to ensure high performance and low latency.

III. METHODOLOGIES

- Dataset creation: Images of a single pupil will be taken from various angles and motions. These pictures are pre-processed. The photographs are cropped to provide the Region of Interest (ROI) that will be used in the recognition procedure. The following step is to resize the cropped image to a specific pixel so that it can capture These pictures will be changed from RGB to grayscale, and they'll be saved in a folder with the student's name on it.
- Face Detection: Use of the hear-cascade classifier with open cv is made for face detection. Before the system is properly trained to detect faces, we are utilizing the hear-cascade algorithm to detect human faces. It is known as future extraction. the xml file for hear-cascade -frontal face -default uses the hear-cascade training data. The extracted features will be derived from the detected features.
- Hair Features: Right now, we are utilizing open cv to detect multiscale. When the image is recognized, a rectangle will be drawn around the faces. The three crucial criteria to take into account are size, scale factor, and minimum neighbors Scale-factor It details the amount by which the image should be reduced. each image's scale Min-neighbors: The bare minimum number of neighbors that each candidate Rectangle may have is defined here. Fewer faces are identified by higher values, but the image quality is better. The smallest object size is identified by Min Size. It starts off at (30, 30). Scale Factor is one of the parameters in this system.
- Face Recognition: Three steps make up the face

recognition process: prepare training data, train face recognizers, and make predictions. Here, the photographs in the dataset that make up the training data will be the ones used to identify faces. These images will be assigned in the students' tables as belonging to those images. The system uses a local binary pattern histogram as the face recognizer. First, the list of local binary patterns (LBP) for the complete face is created. These LBP are decimalized, and then histograms of each decimal value are produced. Each image in the training data will eventually have a unique histogram built for it. In the future, during the recognition process, the histogram of the target face is generated., and it is compared with the histograms that have already been computed to return the label that is most closely connected with the student to which the face belongs.

- Attendance Marking: Following completion of the aforementioned face detection and identification procedures, the stage is used to record attendance. The excel sheet will record recognized faces as present and designate the other faces as absent.
- Attendance Updating: This is the final phase, when an updated attendance report will be completed on an Excel spreadsheet. The list of absentees will be mailed to the appropriate faculties. Recognized faces will be recorded as present, and the others will be marked as absent. At the conclusion of each month, faculties will receive an update with their monthly attendance sheet.
- FlowFeatures: - There are several special features that you could include in a research paper on face recognitionwith real-time database attendance using artificial intelligence (AI) and machine learning (ML), using Firebase database. Here are a few potential options:
- Integration with Firebase: One of the key features of your system is likely to be its integration with Firebase, which is a cloud-based real-time database that can be used to store and retrieve data in real-time. You could describe how your system interfaces with Firebase, how data is stored and retrieved, and how the database is used to track attendance.
- Real-time attendance tracking: Another key feature of your system is likely to be its ability to track attendance in real-time. You could describe how the system uses face recognition to identify individuals as they enter a room, and how this data is used to update the attendance database in real- time.
- Accuracy and performance: One of the most important aspects of your system will be its accuracy and performance. You could describe how you trained your machine learning models to achieve high accuracy rates, and how you optimized your system for speed and efficiency.
- User interface: Another key feature of your system is likely to be its user interface. You could describe how you designed the user interface to be easy to use, intuitive, and responsive, and how you integrated it with Firebase to provide real-time feedback to users.
- Security and privacy: As with any system that uses biometric data, security and privacy are key concerns. You could describe how you designed your system to protect user data, including measures such as encryption, access control, and auditing.
- Overall, these are just a few potential special features that you could include in a research paper on face recognition with real-time database attendance using AI and ML, using Firebase database. Depending on the scope of your research, there may be other features that are relevant to your study as well.

- Flow chart:

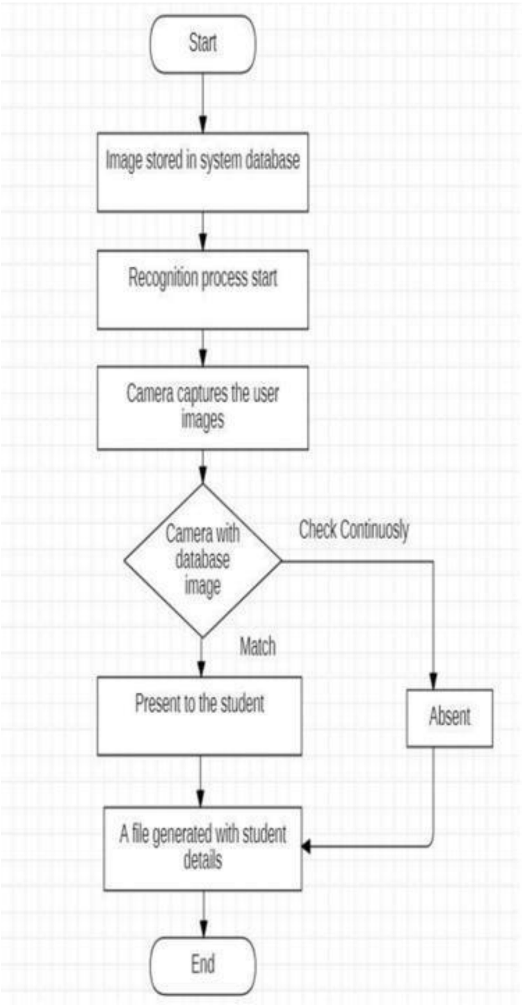
Start
Image stored in system database
Recognition process start
Camera captures the user images
Camera with database image
Check Continuosly
Match
Present to the student
Absent
A file generated with student details
End

Figure 1: Flow chart of Algorithm

- *Algorithm Used:*

AdaBoost (short for "Adaptive Boosting") is a popular machine learning algorithm widely used for facial recognition. In AdaBoost, several "weak" classifiers are combined to create a "strong" classifier that can classify faces accurately. AdaBoost is used for facial recognition as follows:

A. Feature Extraction: The first step is to separate the facial image's features. The placement and density of specific facial features like the eyes, nose, and mouth as well as numerous aspects like Gabor filters or Haar-like features are examples of these features.

B. Training Weak Classifiers: Then, various weak classifiers are trained using feature extraction. These weak classifiers are usually simple decision trees or other classifiers trained on training data.

C. Weighting Data: With each iteration of AdaBoost, the training data is weighted so that false samples are further weighted. This means that the next weak classifier will mostly focus on previously classified samples.

D. Combining Classifiers: Strong classifiers are created by combining all trained weak classifiers. This is done by weighting each weak classifier according to its accuracy and then combining its predictions using a weighted voting algorithm.

E. Face Recognition: Finally, the face in the new image is recognized using a strong classifier. The device displays a confidence score indicating whether the image has a face, and if the confidence score exceeds the threshold, the image is classified as a face. The AdaBoost has proven to be a good facial recognition system with many practical applications. However, it is important to remember that AdaBoost can be expensive, especially when using large sets or a large number of parameters.

- *Hardware requirements*

A. Arduino Uno Board: An Arduino Uno, a development board for microcontrollers based on the ATmega328P microcontroller, is required.

B. Raspberry Pi Camera Module: Compatible with Raspberry Pi single-board computers, the camera module has two versions, one with an 8-megapixel sensor and the other with a 12- megapixel sensor. It is relatively inexpensive and delivers high-quality images in a compact format.

C. USB Cable: The Arduino board is connected to your computer using a USB wire. It is necessary to use a USB cable with a USB type A connector on one end and a USB type B connector on the other.

Flow and Processing of Algorithm:

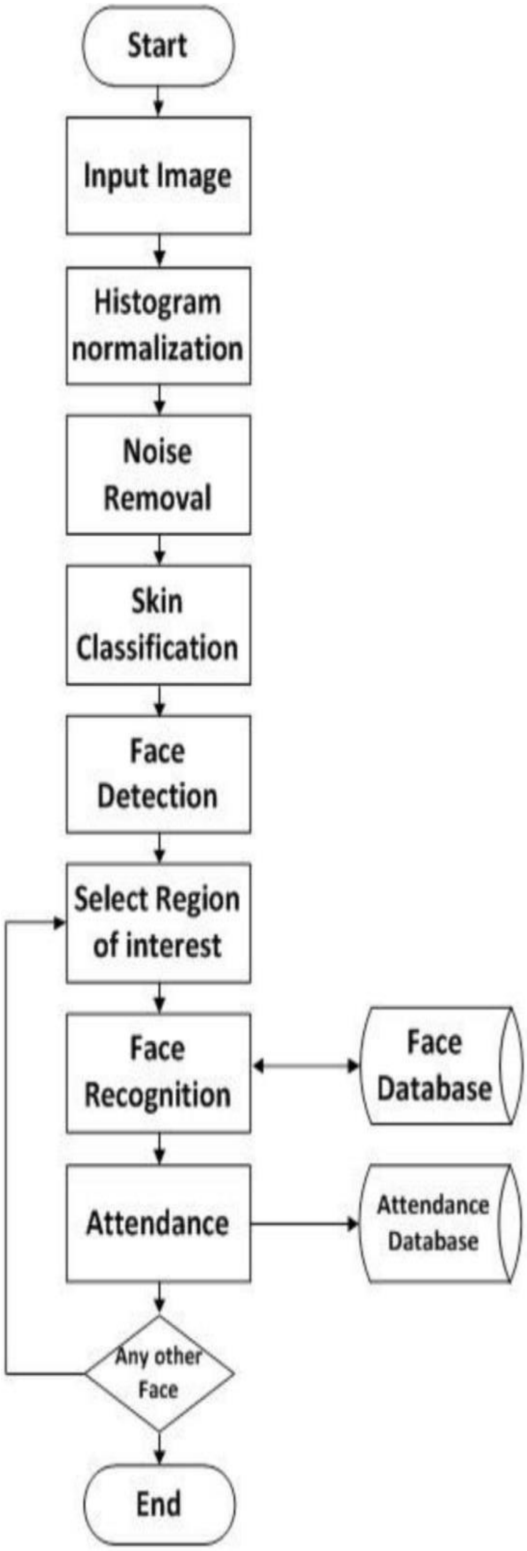

Figure2: Flow and process of algorithm

IV. RESULT AND ANALYSIS

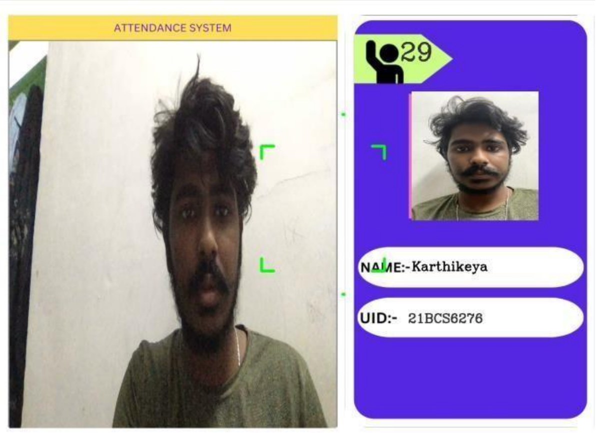

Figure 2: Extracting Information

Step1: It will recognize the face and extract the data from the database and show this graphic design the details showis name, uid, and how many classes you are present and image of the student

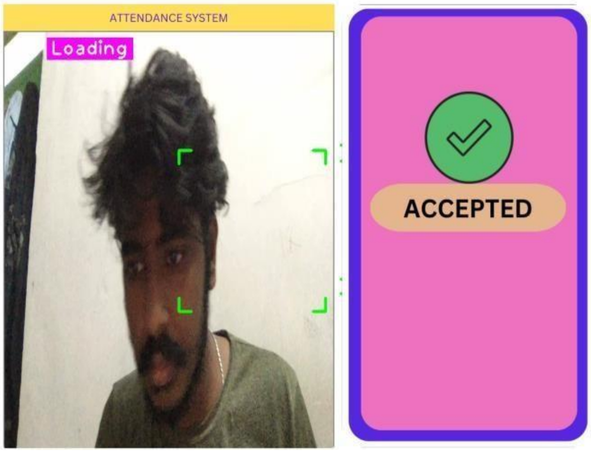

Figure 3: marking attendance in real time

Step2: After recognizing it from database and it will mark the attendance in real time in firebase database so it is easy to access and manage the data

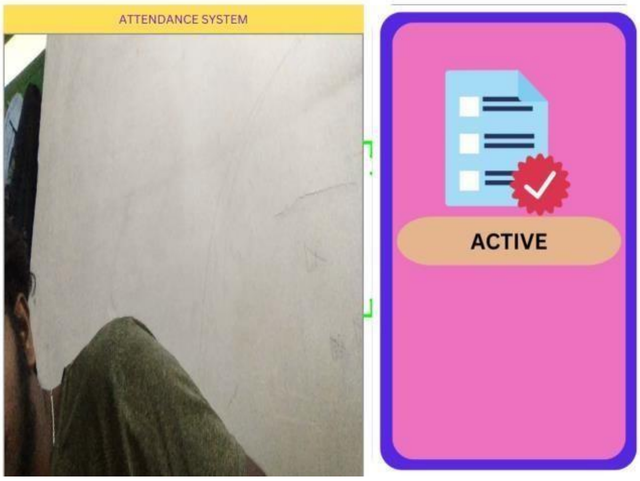

Figure 4: state of student

Step3: After completion of these process when student comes before the camera who is already marked it will be accepted and he is the condition of active.

REFERENCES

[1]. Georghiades, A.S., Belhumeur, P.N. and Kriegman, D.J., 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6), pp.643-660.

[2]. Viola, P. and Jones, M., 2001. Rapid object detection using a boosted cascade of simple features. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (Vol. 1, pp. I-511). IEEE

[3]. Yang, M., Kriegman, D. and Ahuja, N., 2002. Detecting faces in images: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(1), pp.34-58.

[4]. 4. Turaga, P., Chellappa, R., Subrahmanian, V.S. and Udrea, O., 2008. Machine recognition of human faces. Springer Science & Business Media.

[5]. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J. and Tyson, J.G., 2000. The FERET evaluation methodology for face recognition algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(10), pp.1090-1104.

[6]. Moghaddam, B., Pentland, A. and Srihari, S.N., 2000. A bayesian framework for face recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (Vol. 1, pp. 176-183). IEEE.

[1]. Zhao, W., Chellappa, R., Phillips, P.J. and Rosenfeld, A., 2003. Face recognition: A literature survey. ACM Computing Surveys (CSUR), 35(4), pp.399-458.

[2]. Wang, Y., Huang, T. and Wu, G., 2004. Face recognition using discriminant locality preserving projections. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (Vol. 2, pp. II-209). IEEE

[3]. Tan, X. and Triggs, B., 2010. Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE Transactions on Image Processing, 19(6), pp.1635-1650.

[4]. Hu, J., Lu, J., Tan, Y.P. and Zhou, J., 2014. Discriminative deep metric learning for face verification in the wild. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 1875-1882). IEEE