Design of Computer Vision-Based Attendance in School Setting

Prof.Mrunalini Buradkar
Department of Electronics & Tele-communication
St.Vincent Pallotti College Of Engineering & Technology
Nagpur, India
Email Id-*mburadkar@stvincentngp.edu.in*

Abhilasha Vinod Atey
Department of Electronics & Tele-communication
St.Vincent Pallotti College Of Engineering & Technology
Nagpur, India
abhilashaatey25@gmail.com

Shreya Kamble
Department of Electronics & Tele-communication
St.Vincent Pallotti College Of Engineering &Technology
Nagpur, India
kambleshreya422@gmail.com

Unnati Rathod
Department of Electronics & Tele-communication
St.Vincent Pallotti College Of Engineering & Technology
Nagpur, India
Unnatirathod1401@gmail.com

Arsheel Sheikh
Department of Electronics & Tele-communication
St.Vincent Pallotti College Of Engineering & Technology
Nagpur, India

Sheikharsheel@gmail.com

Mohit Ravi
Department of Electronics & Tele-communication
St. Vincent Pallotti College Of Engineering & Technology
Nagpur, India
Mohitravi162@gmail.com

Mohit Pande
Department of Electronics & Tele-communication
St.Vincent Pallotti College Of Engineering & Technology
Nagpur, India
Pandemohit77@gmail.com

Abstract— *Technology growth is speedy, and more and more things can be solved easily with the existence of sophisticated technology. One such technologies is helping to solve the issue of attendance by students at schools. The goal is to provide a tool that reduces the time teachers spend taking attendance while also collecting data that improves teaching practice. The project was created to help teachers save time as well as inspire students and instructors to become more knowledgeable about computer technology, science, and machine learning when they implement and alter the software in their personal classes. In India, we are facing more problems in taking attendance, by writing form it takes more time to take a huge number of students. In this paper, we are presenting face recognition which can overcome the previous problem in India because it's only using faces for attendance. This technology is mainly for school and college students and faculty as well.*

Keywords – *Technology, Student, Computer Vision, Face – Recognition, Attendance.*

Introduction

Computer Vision (CV) is part of Artificial Intelligence (AI) which assists computers understand the real world. Computers with the help of digital images and deep learning models can detect objects and humans and react to them. Computer vision will allow schools to electronically record attendance by together a facial recognition application. In addition to saving teachers valuable time, it can also assist in the education of teachers and students on coding as well as the wide range of possibilities for applications through computer vision. Cameras can be used to detect the object or image for analysis. The image captured is forwarded to the interpreting device which is used to break the captured image and compare the patterns present in the image with the libraries present in it and check whether the patterns match with the image. The patterns can be anything like how the object looks and the appearance of the

object. Face recognition can be described as a technique which can identify a person's face through video footage using a database of facial features that is typically used for identification purposes through identity verification. In this project, we have used Deep Learning Library (dlib) which is a C++ library used for implementing machine learning algorithms and has face recognition capabilities, facial landmark detection, and is used for creating attendance systems.

Facial landmark detection is used to detect key points on the face like eyes, nose, mouth, etc. dlib generates 128 – 128- dimensional face descriptors using a deep convolution neural network (CNN). By that, these words create a sketch of the face, determining the individuality of a person. the facial recognition dlib is based on comparing the face descriptor of one individual with the data that is in the database to be able to identify the features of the person. This comparison involves the "Euclidean distances" between descriptors, which are the key characteristics of the face. Therefore, this face recognition algorithm becomes a very important factor in the attendance monitoring process for the students.

As part of the application, we have also used "Flask" which is a WSGI web application framework written in Python programming language for creating the web application interface for viewing attendance records. The libraries used in the project include "Open CV" which is the library for capturing images from the camera processing them and doing operations like face detection. "Numpy" for adding large, multi-dimensional arrays and matrices. "Pandas" to handle data storage for extracting features. "SQLite" is a C language library that executes a small, fast, full-featured SQL database engine and it is used for storing attendance records. "Scikit-learn" is a collection of algorithms for processing images in Python.
The hardware components used in our project are "Raspberry Pi" it is a small, affordable computer capable of running the face recognition system. "Webcam" which is used to capture live video for detecting and recognizing faces. "Mac/PC/Linux" system to run the program which can provide higher FPS for face recognition.

LITERATURE SURVEY

[1]The purpose of this research project is to create a facial recognition-based attendance control system. In the majority of educational institutions, a faculty member in a classroom manually records each student's attendance using an attendance sheet. The goal of this project is to reduce the amount of time spent tracking students' attendance and address the proxy issue by nearly doing away with the need for paper records during the attendance process. Attendance that is manually recorded is easily altered and influenced. Additionally, manually verifying attendance is not a simple task. Thus, we are putting forth a technique that, by utilizing all of the facial recognition data, can more accurately predict the attendance.
[2] Develop an unobtrusive, automated attendance management system that is unobtrusive and has very high degree of accuracy in face recognition

algorithms and face detection for all pose variations together the highest resolution camera, without the intervention of a human.
The proposed system comprises an ultra-high definition rotating camera that is placed inside the class to record all students. This project uses the high-definition camera will be placed inside the classroom in such it is so that the view from the front of the camera encompasses a wide range from 70 to 80% whole class.[3] Additionally, Python was selected as the principal software language in the project because it's highly readable and is easily accessible to intermediate and novice developers. The language we used was OpenSource Computer Vision Library (Open CV) which is widely utilized for facial recognition. In this project, we had also used Deep Learning library(dilib) of C++ which are known for its comprehensive implementation of machine learning algorithms, especially for facial recognition.[4] Dilib is an important library in identifying key points on face such as eyes, nose, mouth and jawline. We had also used 'Flask' which is WSGI web application framework in python used to create web application interface for viewing attendance Libraries of Python such as Numpy and Pandas are used to to collect and store multi-dimensional arrays and matrices and also to handle large data sets for feature extraction and to process the data respectively.

The hardware components used in this project is Raspberry Pi which is a small affordable computer used to carry out tasks related to facial recognition. Also, a web-cam is used in the project to record the faces of the individuals.
So, with the help of this project, manipulation with attendance of students and the time which used to get wasted during traditional system would be saved.

Methodology

Software-To properly be able to execute this project we will need the code for face detection and face recognition. This highlights the need of integration of hardware (Raspberry Pi and camera) and software (python environment with deep learning libraries). For this, firstly we need software and framework setup. This done by setting up an operating system for Raspberry Pi, installing deep learning and machine learning libraries (Dlib, OpenCV), a database for storing attendance records, and flask for creating a front –

end interface. In this project, we use a pre-trained CNN (convolutional neural network) model,like Dlib face recognition model , which is based on deep learning techniques to extract 128D facial descriptions and machine learning models for comparing facial features . This system uses the k- nearest neighbour algorithm to find the best match for the detected face.

The enrollment process of a person's face is done by capturing multiple images of a person's face through different angles. This is done by using CNN-based Dlib's get _frontal_face_detector. Uniformity in facial feature extraction is ensured by extracting and aligning facial regions. The captured image is then saved in the database along with the person's name. Processing of the saved face

image is done through the CNN model to generate 128D facial descriptors. The extracted descriptors is stored along with the person's name in the database, creating a unique identifier for each enrolled person.
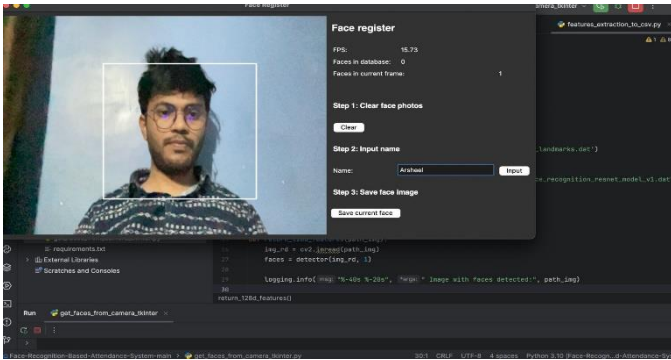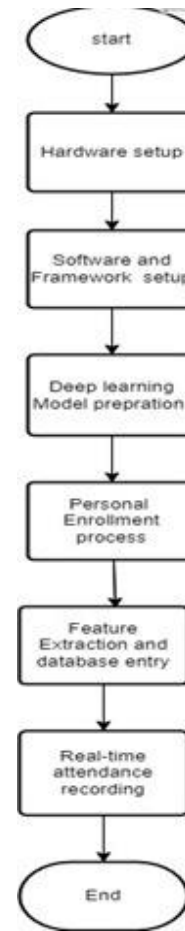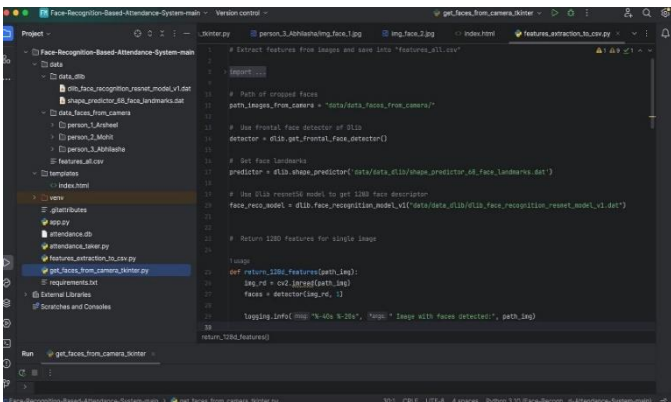

Fig- Registering face from camera

For students to record live attendance, we use the CNN model to detect the face. Machine learning algorithms are used to compare the live feed descriptors with those in the database to identify the person. If a match is found the identified person's name and timestamp is stored in the database. Flask web application serves as the front -end interface which helps the students with attendance verification and to interact with the system . This system also allows users to input a data and retrieve attendance data for that date from the database. The working of this system can be more easily described through the flow chart and block diagram -
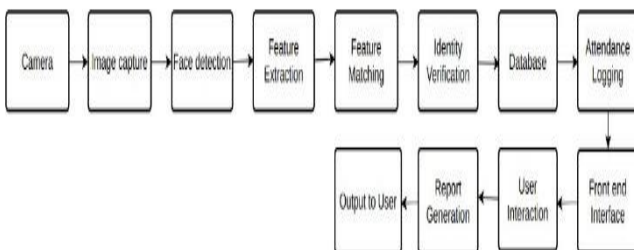
Flowchart -

Hardware -Hardware components –

1. Raspberry Pi: A small, affordable computer capable of running the face recognition system. The project notes the system's compatibility with Raspberry Pi, demonstrating its efficiency and low resource requirements.
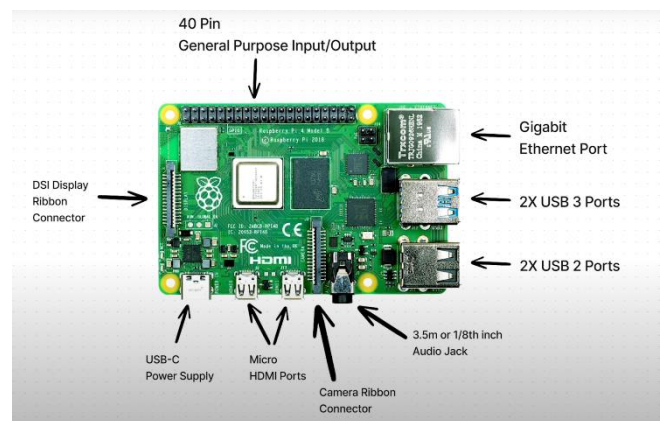
2. Webcam (HPW 300 or any compatible webcam): Used to capture live video feed for face detection and recognition. The system is adaptable with various webcams, ensuring flexibility in hardware choices.

3. Mac/PC/Linux System : The project also details running the system on more powerful hardware like a Mac, PC, or Linux system, which can provide higher frames per second (FPS) for face recognition.

Setting Up Raspberry Pi Heedlessly on laptop: A Step-by-Step Guide

Tools Needed:

1. Micro SD card (8 GB or more)
2. Computer with a microSD card slot or an external card reader
3. Raspberry Pi power supply (2.5A, 5V) with a micro-USB or USB-C cable
4. Raspberry Pi Imager software
Step 1: Download the Raspberry Pi Imager Software

- Plug the microSD card into your computer or card reader.
- Download the Raspberry Pi Imager from [raspberrypi.com] (https://raspberrypi.com).
- Install the Imager following the on-screen instructions.

Step 2: Use the Imager to Install Raspberry Pi OS
- Open the Raspberry Pi Imager.
- Choose "Raspberry Pi OS (64-bit)" as the operating system.
- Select your microSD card.
- Access the "Advanced Setup Menu" by clicking the gear icon.
- Configure Wi-Fi settings, hostname, enable SSH, and set a password.
- Save the settings and click "WRITE."
Step 3: Insert the microSD Card and Power Up
 - Insert the microSD card into your Raspberry Pi.

- Connect the power supply to the micro-USB or USB-C power connector.
- Wait for the Raspberry Pi to boot (this may take up to 30 seconds).
s
Step 4: Generate an SSH Key

- Open Terminal on your laptop.
- Copy and paste the following line to allow your laptop to connect to the Raspberry Pi:

```
ssh-keygen -R raspberrypi.local
```

Step 5: Log into Your Raspberry Pi

- Use the following command to log in, replacing "hostname" with your chosen hostname:

```
ssh pi@hostname.local
```


- Follow prompts to confirm connections and enter your Raspberry Pi password.
Step 6: Update the Software on Your Pi

- In Terminal, run the following commands:

```
sudo apt-get update -y
sudo apt-get upgrade -y
```

- Confirm disk storage usage during the upgrade process.
Step 7: Proper Shutdown

- To shut down your Raspberry Pi safely, use:

```
sudo halt
```

Accessing Raspberry Pi Desktop on Mac without a Keyboard or Mouse: Headless VNC Setup

Step 1: Setting Up Your Raspberry Pi

Before diving into VNC setup, ensure your Raspberry Pi is connected to the same Wi-Fi network as your computer. If you haven't set up your Raspberry Pi yet.

Step 2: Installing VNC Viewer
Start by downloading the VNC Viewer software from the RealVNC website. Navigate to their website and click on the "Download VNC Viewer" button. Follow the prompts to save the file, typically to your Downloads folder. Once downloaded, locate the file and open it. Drag the VNC Viewer into your Applications folder to complete the installation.

Step 3: Configuring Raspberry Pi

1. Open Terminal

2. Use SSH to log into your Raspberry Pi. Enter the following command, replacing "hostname" with your Pi's hostname:

```bash
ssh pi@hostname.local
```

3. Enter your Pi's password when prompted.
4. Launch the Raspberry Pi configuration tool by entering:

```bash
sudo raspi-config
```


5. Navigate to "Interfacing Options" and select "VNC." Enable VNC by selecting "Yes" and pressing Enter.
6. Configure display options (e.g., resolution) as desired.
7. Reboot your Raspberry Pi to apply the changes:

```bash
sudo reboot
```

Step 4: Finding Your Pi's IP Address

After rebooting, log back into your Raspberry Pi using SSH. Use the following command to retrieve your Pi's IP address:

```bash
ifconfig
```

Step 5: Connecting via VNC Viewer

1. Launch VNC Viewer on your laptop.
2. Paste your Raspberry Pi's IP address into the address field and press Enter.
3. When prompted, enter your Pi's username (usually "pi") and password.
4. Click "Remember my password" for convenience and click "OK" to log in.

Step 6: Optimizing the Experience

- Adjust desktop preferences: Right-click on the desktop, select "Desktop Preferences," and choose a simpler background image or none at all to improve performance.
- Explore the desktop: Navigate through applications, files, and settings just like you would on a local desktop.

Step 7: Shutting Down the Pi

When you're done using your Raspberry Pi, shut it down properly by clicking on the Raspberry icon, selecting "Shutdown," and clicking "Shutdown" again.

How to run it on raspberry pi

Setting up the Face Recognition System on pi

1. Cloning Repository: To begin, clone the GitHub repository with the system's code into your Raspberry Pi device. This process can be achieved by running 'git clone' with the specified URL of the repository.

2. Data Preparation: Make sure that all important data files are downloaded to the Raspberry Pi. This can be achieved with the use of FileZilla for seamless file transfer, or get the involved files directly from the repository.

3. Installation: Install the OpenCV library and dlib essential libraries on the Raspberry Pi. These libraries contributes to the collection of essential tools for facial recognition and image processing. A 64-bit Raspberry Pi version such as Bullseye provides better performance and will make the installation running more seamlessly.

4. Running the System: Run the principal system files, 'git_faces.py' as well as `feature_extraction.py` to set up the face recognition system. These files involve the sourcing of facial features and exporting the needed CSV files for the latter processing step.

5. Testing: Take a face-shot with the connected camera and run a series of tests to guarantee the system's quality. Make certain the system correctly identifies people's faces and plots their attendance based on the recognized faces.

Conclusion

The conclusion of an attendance system based on computer vision used in schools would normally summarise the most important results and findings of the research. Here's a sample conclusion: The prototype of the vision-based attendance system in a school environment concludes that it has a large capability of succeeding in increasing the efficiency and effectiveness of the monitoring of student attendance and well-being. Such a system can greatly diminish the human efforts for monitoring attendance and also bring out the facts like the students' activities in school. Consequently, utilizing computer vision technology enables us to automatize some of the mundane administrative tasks and take a more comprehensive viewpoint regarding the learners' requirements. But, it's crucial to remember that the performance of this system is contingent upon proper data privacy and security measures in order to safeguard data of students. With the technology still in its early stages, much work needs to be done to improve the system and to address any issues that might occur as it progresses. In the end this ground breaking technique may well offer the potential to have an influence on the school environment, creating a more caring and attentive climate.
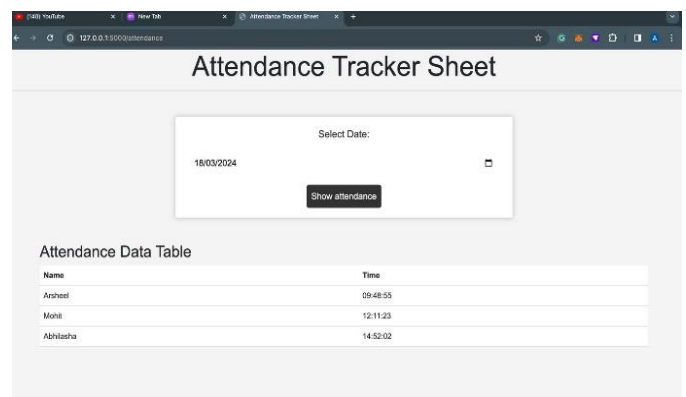

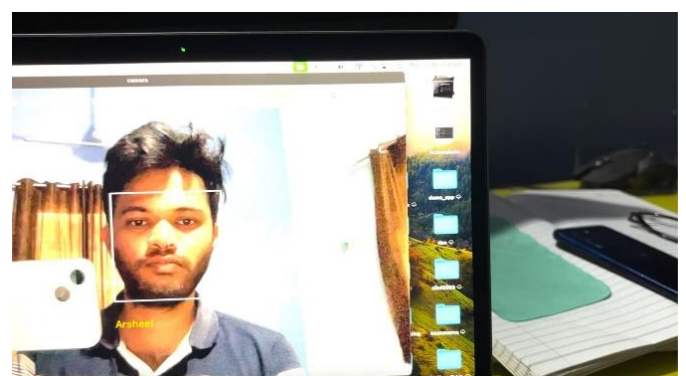*Fig – Real time attendance tracker sheet on flask*

Fig- Attendance Taker

REFERENCES

[1] *Sarah Deniz and Dakyung Lee, "Computer Vision for Attendance and Emotion Analysis in School Settings" IEEE Conference publication 2019.*

[2] *Ngo Tung Son and Tran Quy Ban, "Implementing System Using Deep Face Recognition. "A Case Study at FPT Polytechnic College 2020.*

[3] *Bojun Zou, "Design of the Student Attendance Management System Based on the Internet of Things technology" Volume 2022 Article*

a. *ID 1990530 https://doi.org/10.1155/2022/1990530.*

[4] *Florian Schroff and James Philbin, "Face-Net: A unified Embedding for Face Recognition and Clustering. "Google Inc.17 June 2015".*