Face Recognition Based Attendance System

Rishi Sharma (B17138)*, Rohit Agarwal (B17100)†, Saurabh Bansal (B17059)‡
Shaurya Pratap Singh Tomar (B17102)§, Swapnil Rustagi (B17104)¶ and Varun Singh (B17110)‖

Email:(*B17138, †B17100, ‡B17059, §B17102, ¶B17104,‖ B17110)@students.iitmandi.ac.in

Instructors: Dr. Hitesh Shrimali, Dr. Gopi Shrikanth Reddy

***Abstract*—Our project aims to make the lives of teachers easier by automatically marking students' attendance using Face recognition. It'll reduce the time wasted in taking attendance and also prevent proxy attendance. This project will also upload the attendance directly to a spreadsheet. It will not cause any disturbance during teaching and also save time. Although, it has some limitations on the distance from the students from which it can work, the use of high quality cameras and multiple cameras can reduce it. The project is quite realistic and efficient.**

***Index Terms*—Face Recognition, Attendance, Raspberry Pi**

I. Introduction

In large classrooms, taking attendance is a time-consuming task.

Manually marking attendance is time consuming and to prevent proxy attendance teaching assistants are required to personally monitor the process in case of large classrooms with many students.

Alternate methods of attendance like scanning ID cards, Fingerprint based attendance system are also not practical since every student needs to go to the device and mark attendance, which is time taking. Also there is an issue of proxy attendance, as someone can leave the class right after marking attendance.

This project aims to make attendance process easier by reducing time required, and also making sure that there is no proxy attendance. The project relies on face recognition. Using a high quality image of the classroom, the software can recognize faces and marks the attendance in a spreadsheet. This process can be done while the class is going on, without causing any disruption, and done multiple times to ensure nobody can get their attendance marked without attending the entire class.

II. Components Used

S.No	Component	Quantity	Approx. Cost
1	Raspberry Pi 3	1	3000
2	Logitech C270 Webcam	1	1200
3	16 GB SD Card	1	450
4	L298N Motor Driver	1	300
5	200rpm Motor	1	250
6	Jumper Wires	$\sim$ 10	4 per piece
7	9V Battery	1	25 per piece

III. Description of Components

- **Raspberry Pi 3** - Raspberry Pi 3 is a single-board computer i.e. it is a computer to which input and output devices can be connected for usage and provides GPIO pins, USB ports, HDMI port, Ethernet port etc. It is the main component in our project, used to run all the scripts and libraries for face recognition, recording the attendance, giving input instructions to motor driver etc.

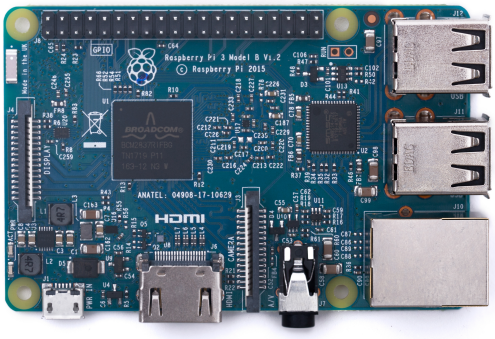

Fig. 1. Raspberry Pi

- **Logitech C270 Webcam** - Used in the project for capturing the photos of the classroom or for taking images to add students to database.

Fig. 2. Camera

- **SD card** - It is the secondary storage device for Raspberry Pi 3.

- **L298N Motor Driver** - The motor driver is used to control the motor based upon GPIO pin output from Raspberry Pi in this project. Thus, our program being executed on Raspberry Pi can simply change GPIO output to rotate the motor in the direction we want.

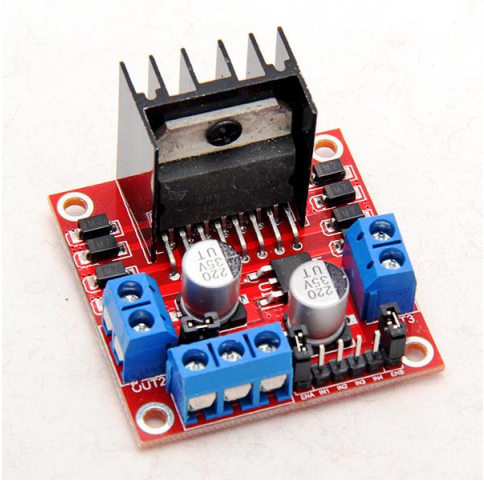

Fig. 3. Motor Driver

- **200 rpm Motor** - Used to rotate the camera so that a wider area of the class can be captured.
- **Jumper Wires** - Used for connections between Raspberry Pi and Motor Driver.
- **9V Battery** - Power source for motor.

IV. Libraries Used

- **Microsoft Azure Face API** - Used to add faces to a person's ID and detect them. Limited trial version of the library is used. Server connection is required for image processing.
  Version == 1.4.1
- **dlib** - Used to detect faces from a given picture and crop them out from original image.
  Version == 19.10.0
- **OpenCV** - Used to take pictures from camera and store it.
  Version == 3.4.0.12
- **Python** - Version == 3.6.4
- **Openpyxl** - Used to store data in a spreadsheet from a python file
  Version == 2.5.3

V. Working

The circuit diagram is as shown below

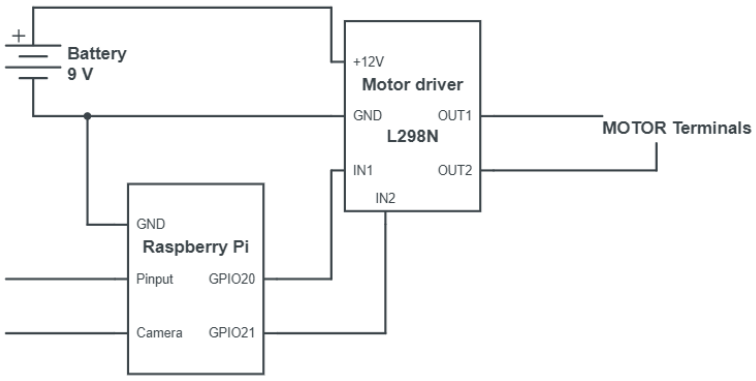

Fig. 4. Circuit Schematic

Before using the attendance system, a database of students whose attendance is to be recorded has to be created. To add students to the database, a student's name and roll number has to be entered and then 20 photos of the student will be taken by the camera connected to the Raspberry Pi. The student should try to ensure that these 20 photos comprise of different face angles, this will improve the accuracy of the face recognition. Now, the face database is trained and is ready to use.

Upon starting the attendance script on the Raspberry Pi, the script takes a photo from the camera. Then face detection is run on this photo. The faces are cropped from the photo and stored. Then face recognition is run on these cropped faces and attendance is marked in a spreadsheet for all recognized students. The script automatically marks the attendance in the column corresponding to today's date. Then the motor rotates the camera and another photo is taken.

This process is repeated is repeated such that 2 photos are clicked when camera is straight facing, and 2 photos each are clicked when camera is rotated at an angle.

VI. Procedure

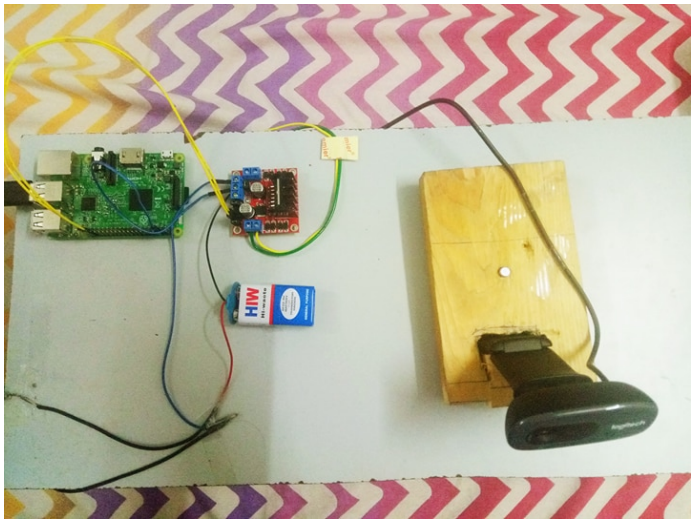

Fig. 5. Camera

1) Write the code to store data of a new entry using their name and roll number.
2) Make another script to automatically mark attendance in a spreadsheet.
3) Transfer the code to Raspberry pi and connect the components according to the given circuit schematic using jumper wires.
4) Mount the camera on top of the motor and place other components like raspberry pi, motor driver, etc. on the stand.
5) Record names, roll number and pictures for the dataset and train the database.
6) Setup the camera in Raspberry pi and run the script for marking attendance automatically whenever required.

VII. Directions for use

1) To add a new record:
   Execute the file add_student.py and input the name and roll number of the student. Now run the files create_person.py and add_person_faces.py to add the faces to the profile. Now, train the database using train.py file.
2) To mark attendance:
   Execute the master.py file in main directory to take start the camera and motor which will then automatically take pictures of the whole room and mark the attendance.

VIII. Advantages

1) Microsoft Face API is highly accurate and disregards changing facial features such as beard and moustache and also detects the person with or without spectacles.
2) The maintanance cost shall be quite low, as there as no wear and tear.
3) This system can give real time update of attendance to the students and the professors, which shall be quite helpful to both.

IX. Shortcomings

1) If Microsoft Face API is used, a permanent internet connection with a subscription to the service is required. A free and offline library like face_recognition can be used as an alternative.
2) A good quality camera or multiple cameras are required for larger classes.
3) The results are susceptible to improper lighting conditions in the room.

X. Observations

Dataset was made for all group members and attendance was marked using a test picture containing all group members excluding Saurabh Bansal. The results obtained are also shown:

	A	B	C	D
1	Roll Number	Name		
2				
3	B17100	Rohit		
4	B17102	Shaurya		
5	B17104	Swapnil		
6	B17110	Varun Singh		
7	B17138	Rishi Sharma		
8	B17059	Saurabh		
9				
10				

Fig. 6. Spreadsheet without attendance

Fig. 7. Test image

	A	B	C	D	E
1	Roll Number	Name	22_05_18		
2					
3	B17100	Rohit	1		
4	B17102	Shaurya	1		
5	B17104	Swapnil	1		
6	B17110	Varun Singh	1		
7	B17138	Rishi Sharma	1		
8	B17059	Saurabh			
9					
10					

Fig. 8. Spreadsheet after marking attendance

XI. Conclusions

We were able to achieve the following:

1) Automatically click multiple images of different portions of the class by rotating camera.
2) Detect and recognize faces accurately from the clicked images through a distance of 5m.
3) Using a better quality phone camera greatly improved the distance of working. One Plus 5 gave a working distance of 10m.

XII. Future Possibilities

We were able to get very close to our objective using the prototype. Using a cheap webcam, we were able to identify all the students in an image captured from a short distance.
Budget utilized: Rs. 5400
To get the attendance accurately for a big class, we need multiple good quality wall mounted cameras. They shall click images of the class for a callibrated duration. All the clicked images shall be processed and attendance shall be marked for the present students. This calls for a high one-time investment but the maintenance cost shall be quite low. Another version of this Automatic Attendance System would be to have a path mounted on the ceiling, over which a high quality camera shall be able to move and click images of different sections of the class, which shall be sent to the processing unit via Wi-Fi or Bluetooth and processed.

XIII. Precautions

1) Try to click images from multiple angles while training the database to get better results.
2) The camera might not get opened in time for clicking pictures so proper error handling is required.
3) Proper connection to internet is to be ensured when using the API.
4) A good quality camera is required for superior results.

References

[1] *dlib Docs [ONLINE]*. URL: http://dlib.net/python/index.html (visited on 05/18/2018).

[2] *Logitech C270 image*. URL: https://www.walmart.com/ip/Logitech-C270-HD-Webcam-720p-Black/17616997 (visited on 05/22/2018).

[3] Mark Lutz. *Learning Python, 5th Edition*. O'Reilly Media, 2013.

[4] malharsk27. *Autoattendance-Cognitive*. URL: https://github.com/malharsk27/Autoattendance-Cognitive (visited on 05/22/2018).

[5] *Microsoft Azure Face API [ONLINE]*. URL: https://docs.microsoft.com/en-in/azure/cognitive-services/face/overview (visited on 05/17/2018).

[6] *Motor driver image*. URL: http://www.ashopbd.com/product/l298n-motor-driver-module (visited on 05/22/2018).

[7] *OPENCV Docs [ONLINE]*. URL: https://docs.opencv.org/2.4/doc/tutorials/tutorials.html (visited on 05/18/2018).

[8] *Python Docs [ONLINE]*. URL: https://docs.python.org/3 (visited on 05/18/2018).

[9] *Raspberry Pi image*. URL: https://www.raspberrypi.org/blog/raspberry-pi-brazil (visited on 05/22/2018).

[10] Rishi Sharma and Varun Singh. *Automatic Attendance System Source Code*. URL: https://github.com/Varunvaruns9/Face-Recognition-attendance-system (visited on 05/22/2018).