International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 07 | July - 2021 ISSN: 2582-3930

ONLINE ATTENDANCE USING FACE RECOGNITION

Ritu Gour, Saloni Kothari, Shivangi Dubey

Department of Electronics, Bharati Vidyapeeth College of Engineering Pune, India

Abstract: **The title suggests that this app is multipurpose since not only does the app take the attendance of a student of a class within a matter of seconds but it also works as database for the teachers to store the students data for later use since the data may not be available from the university database at all times.**
The concept of face recognition is to give a computer system the ability of finding and
recognizing human faces fast and precisely in images or videos.

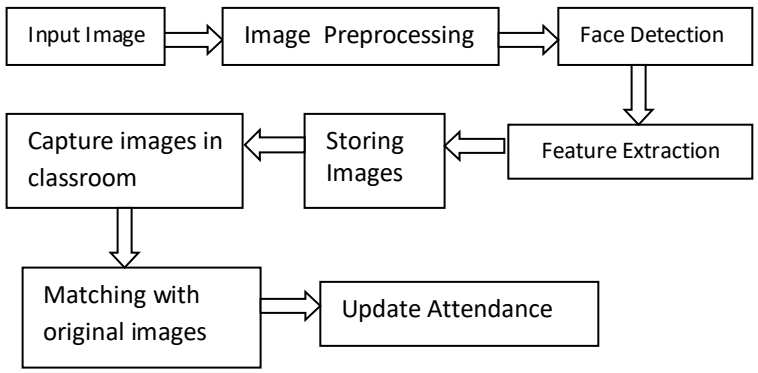
Keywords: ***FaceDetector, Nodejs, MySQL, Microsoft Azure FACE Api , Base64 encoding, Axis IP camera.***

I. INTRODUCTION

In this project, the idea of two technologies namely Student Attendance and Feedback system has been implemented with a face recognition approach. This system automatically detects the student performance and maintains the student's records like their attendance in various subjects. Therefore the attendance of the student can be made available by recognizing the face. On recognizing, the attendance details of the student is obtained as feedback and hence attendance is marked.

II. PROPOSED SYSTEM

The system consists of a camera that captures the images of the classroom and sends it to the image enhancement module. After enhancement the image comes in the Face Detection and Recognition modules and then the attendance is marked on the database server. At the time of enrollment templates of face images of individual students are stored in the Face database. Here all the faces are detected from the input image and the algorithm compares them one by one.



III. MOTIVATION

The motivation behind this proposed work comes from the advancement of technologies like image processing, AI and machine learning and Face unlock feature given by smart-phone manufacturers. The classroom often consists of huge number of students which usually takes a lot of time thus creating a system which will automatically detect the present students and then marking the students accordingly will be very helpful, Just like staff Id scanning but way too easier than that.

IV. METHODS

The image recognition application made by us uses the javascript language.

The teacher's information and detail input panel makes use of the express framework of nodejs to handle the input about the teacher/student details and forward it to the database.

The database is made using SQL with the help of the MySQL workbench tool to create the SQL server.

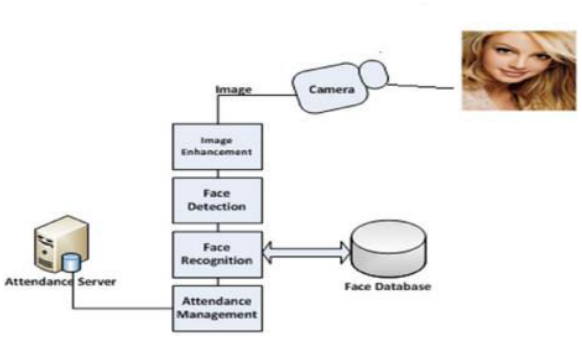
The in-app facial recognition is performed through the FaceDetector library provided by Miccrosoft azure face api.

Frontend: The frontend of the mobile application was made using Javascript and react-native platforms. The front-end of the Teacher information panel was made using HTML-CSS-JS and then converted to Jade View Engine.

Backend: The complete backend and the processes of the teacher input panel were also hosted by the local machine on which they were created (localhost) through nodejs.

It uses the camera module to detect the face through the FaceDetector Library internally. The detected image is then sent to the Microsoft azure servers through the FACE API. The api cannot transport the image as a file thus the image is first cached as a base64 string, then it is decoded using an alternative of the "window.atob()" method to an array buffer. Then a response from the FACE api is expected. According to the response, it is decided whether to proceed forward (on successful detection of face) or loop back to the previous state (on failure). Upon success the student is then asked for the required details of enrolment number and the unique professor identifier so that attendance can be marked. To handle the marking of attendance and other backend requests like updating teacher information on the SQL database, a Nodejs based server is created using expressjs.

V. BLOCK DIAGRAM

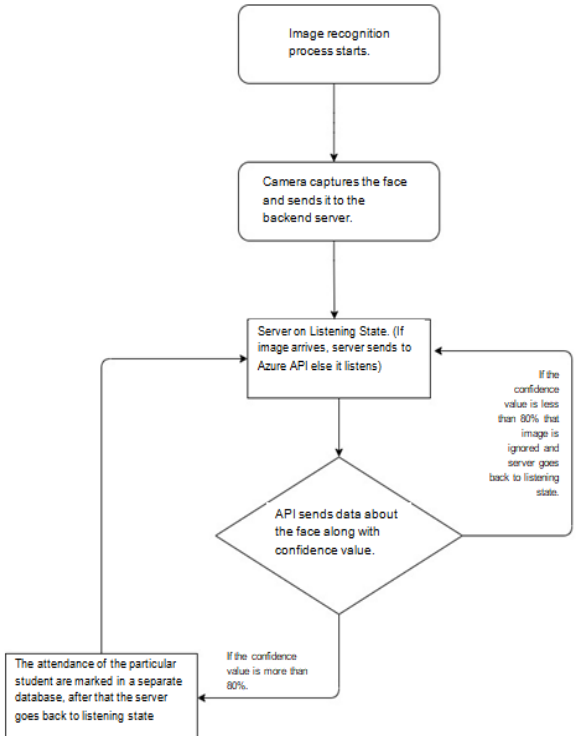


VI. WORKING

This application uses Microsoft Azure face api. The image of student is captured upon entering the classroom. It only proceeds upon verification. It then uses the camera module to detect the face through the FaceDetector Library internally Then a response from the FACE api is expected. According to the response, it is decided whether to proceed forward (on successful detection of face) or loop back to the previous state (on failure). Upon success the student is then asked for the required details of enrolment number and the unique professor identifier so that attendance can be marked. To handle the marking of attendance and other backend requests like updating teacher information on the SQL database, a Nodejs based server is created using expressjs.

The server keeps on listening for a new image.

VII. FLOWCHART



VIII. WORKFLOW:

The face is detected through the camera that is put up outside the class. A student stands infront of the camera. The camera captures the image of the face and sends it to the server. When the server receives the image, it converts the image into base 64 format and sends it to Microsoft Azure face recognition Api to check if the face is present in the database. If the database returns a confidence value of more than 80% the attendance of the student is marked in a separate database. Else, the image is ignored.

IX. RESULT

The face detection response comes directly from the FACE API in the following json format:

```
face detect res:  Array [
  Object {
    "faceId": "abba62d2-1c91-4d24-a41b-4ec4e52a405e",
    "faceRectangle": Object {
      "height": 2308,
      "left": 657,
      "top": 2115,
      "width": 1704,
    },
  },
]
```

This is an Array of detected faces in the api.

***faceId*:** It is the unique id given to every face.

***faceRectangle*:** Details about the shape of the face.

The faces with these details are then then checked if these details exists in the existing face database.

Confidence: This attribute represents the confidence of the detected face. In the figure above, the confidence level is 67.6%.

```
find similars res:  Array [
  Object {
    "confidence": 0.676065743,
    "persistedFaceId": "882e52d2-1132-427d-bb45-6a9a9e614a4e",
  },
]
```

***Persisted FaceId*:** the face id of the face in database with whichit was actually matched.

Out of 13 trials, I got results with:
Confidence below 40%: in 3 trials.
Confidence between 40 80%: in 5 trials.

Confidence above 80%: in 2 trials.

No.	Success/Failure	Confidence	Time (s)	Reason
1	Success	62%	13	-
2	Success	43.5%	12	-
3	Success	73.43%	14	-
4	Failure	0	23	Not Register
5	Success	54.2%	18	-
6	Success	14.34%	11	-
7	Success	82.93%	14	-
8	Success	64.97%	14	-
9	Success	19%	16	-
10	Failure	0	33	Low Light
11	Success	19.33%	21	-
12	Failure	0	31	Low Light
13	Success	91.011%	13	-

Number of successes: 10 / 13, 76.9%

Number of failures: 3/13, 23.1%

Average time for the process to complete (for success): 14.6 s.

The most probable cause of failure: Picture taken in low light.

X. CONCLUSION

Therefore, this application provides an excellent way to evaluate and maintain classroom attendance. This method not only saves paper, but also makes full use of technology to save time in class. This app provides students with their attendance data so that they can track it. Teachers also have more teaching time and can track students' daily attendance**.**

XI. ACKNOWLEDGEMENT

The development of any project requires the opinion of many people, but we suspect that technical projects are different from other projects. Many people participated in all stages of project preparation and shared their knowledge and experiences.
Before we discuss in depth, we would like to say a few heartfelt words to the people involved in this project, who have provided unfailing support since the beginning of stage creativity in many ways.
It is an honor for us to express our deep gratitude and respect to our project guide Mrs. Prajakata Naregalkar. Her initiative, strong interest and expert guidance at every step have provided us with a steady stream of inspiration and encouragement to study this topic in depth. We are deeply grateful to her.
We sincerely thank all those who directly or indirectly help us develop and evolve our ideas.

XII. REFERENCES

[1] https://docs.expo.io/
[2] https://reactnative.dev/docs/getting- started
[3] https://nodejs.org/en/docs/
[4] https://github.com/noble/bleno
[5] https://github.com/mathiasbynens/base64
[6] https://www.npmjs.com/package/axios
[7] https://github.com/axios/axios
[8] Getting Started with the Internet of Things: Connecting Sensors and Microcontrollers to the Cloud, Cuno Pfister, 2011