15EC401M – Multidisciplinary Design

STUDENT ATTENDANCE MONITORING USING FACIAL RECOGNITION IN A CLASSROOM

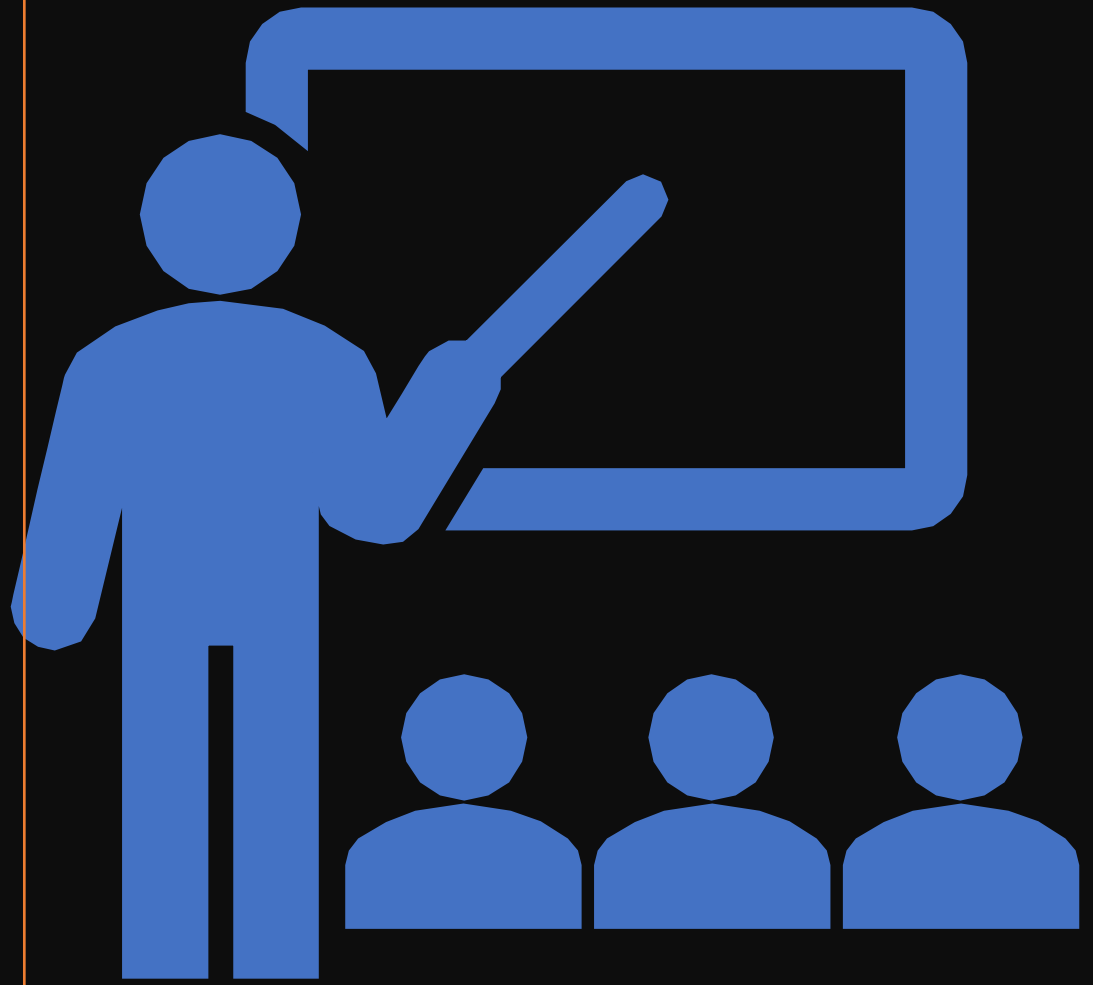
25th Nov 2020

Batch number:18

Team members

1) K CHANDRASEKARAN RA1711004040028

2) S AATHITH RA1711004040030



Existing problem

- Student's attendance in the classroom is a very an important task.
- The existing method of taking attendance consumes a finite of amount of time which can be used for other productive action.
- We are focused on addressing this problem on schools and universities only.

Motivation

We found this problem in many universities and schools.

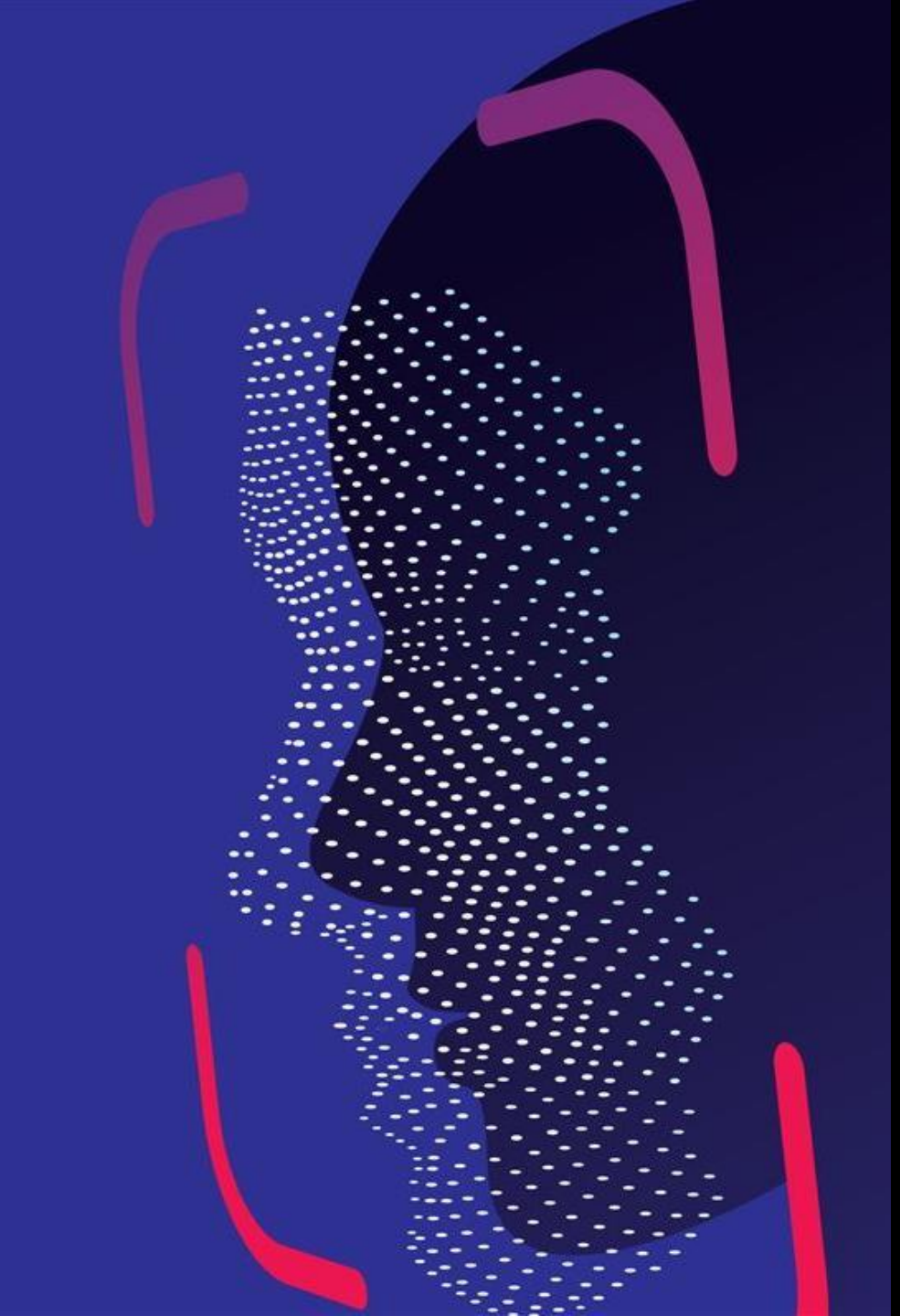
We want a reliable system which identifies which student is present in the classroom and gives the attendance based on his presence.

This system must not be complex and very easy to interact with it.

Literature Survey

Existing Recognition systems

- Fingerprint based recognition system
  - Distracts the attention of students during lecture time
  - Queuing takes place
- RFID based recognition
  - Possibilities that Fraudulent may occur
  - Cheat the system by giving proxy (for example a student misuse it by wearing another student's ID card)
- IRIS based recognition system
  - Can't use a regular camera
  - Visible light must be minimized for maximum accuracy



Proposed work

- Attendance is monitored and marked present or absent depending upon the presence of the student inside the class at the particular time of lecture.
- Our solution uses a camera that takes photo of a group of students, and then uses facial recognition for marking attendance for a student.
- We found that facial recognition overcomes some of the problems that exist in the previously mentioned attendance systems .
- Our solution makes sure that there is a minimal interaction between students and the attendance monitoring system .

Facial recognition

- Any facial recognition has 2 steps
  - Facial detection – the process of finding a face in a given image
  - Facial recognition – the process of identifying the person from the image of a face (example : face id in apple)
- Every 5 minutes the camera takes the picture and facial recognition is done on the picture.
- The process of facial recognition can be implemented in a separate computer as the burden on the camera system must be reduced.

An example of Facial recognition in class

Sub Systems

Major Sub systems

System Attached in the class to take photos (Image processing)

System that marks the attendance against the face in the photos(ML/DL)

System that takes photo

Camera

System that signals the camera

System that transports the photo to next system for processing the photos

System that marks the attendance

Contains the following systems inside it

- System that receives the photo
- System that runs the face detection algorithm that outputs “m” detected faces
- System that outputs a descriptor for “m” faces
- System that iterates through “m” descriptor and marks the attendance

More explanation about a few systems that is inside the sub systems

System that marks the attendance - System that outputs a descriptor for “m” faces

- System that iterates through the “m” detected faces and calls the following sub systems
- System that outputs a descriptor(fixed size) for one face (Facial recognition)
- System that stores these “m” descriptors

System that marks the attendance – System that iterates through “m” descriptors and marks the attendance

- System that iterates through the database containing “n” descriptors
- System that checks whether 2 descriptors are same
- System that gives the register number of a given descriptor
- System that marks the attendance against the register number

Mathematical Modelling

System that takes photo - System that signals the camera

```
output = signal to camera
begin:
    if (5 minutes elapsed):
        signal to camera
        recieve the photo
        store it
        hand over to next sys
```

System that marks the attendance - System that outputs a descriptor(fixed size) for one face (Facial recognition)

```
input = photo-of-face
output = vector(size = d x 1)
begin:
    pre_processing(photo-of-face)
    vector = CNN(photo-of-face-processed)
end
```

System that marks the attendance - System that checks whether 2 descriptors are same

```
n[i] descriptor of size (1 x d) from the database
m[j] descriptor of size (d x 1) from the detected face

input = n[i] and m[j]
output = True or False
begin:
    M = m[j]
    N = n[i]
    theta = MN #cosine similarity
    if (theta > 0.5):
        output = True
    else
        output = False
end
```

Objectives

1st Objective

- One camera
- One person to stand before the camera
- Detect the face in the image and run the face recognition

2nd Objective

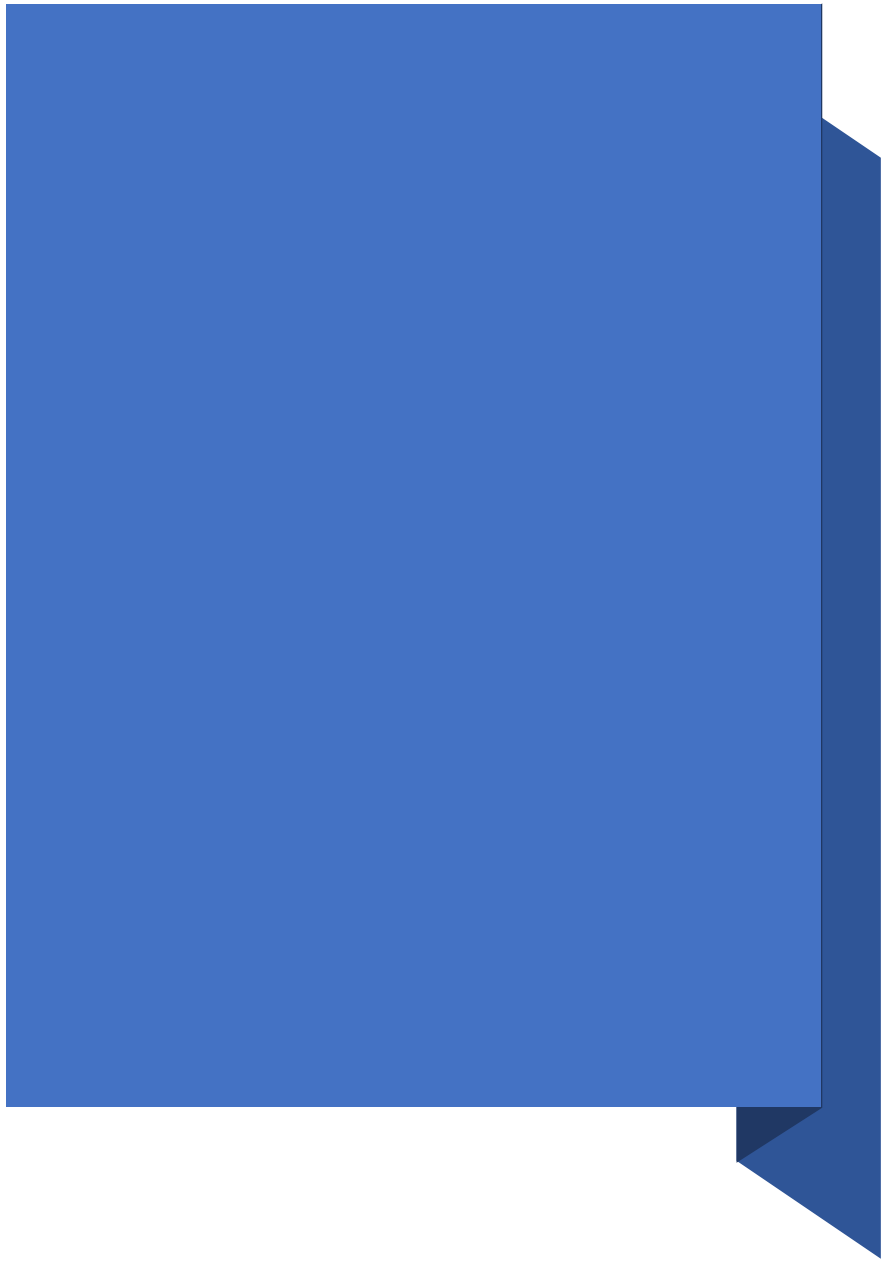
- One person
- A group of people
- Run a face detection model to find the faces in the picture and face recognition algorithm against it

3rd Objective

- Try different face detection algorithms
- Select the algorithm which has lowest false negatives
- Note down other metrics like
    - The least dimension of the facial image that is required for the face detection algorithm.

4th Objective

- Design the camera position , the angle etc.
- The algorithm that was decided on 3rd objective to be used here to test the camera angle and other metrics like finding how many persons can be covered.



5th objective

- Find the best camera angle and algorithm (both face detection and face recognition) that has reduced false negatives, is fast etc.
- Iterate through 3 and 4 objective to find the best combination of camera and algorithm

Constraints

Camera

Proper lighting should be there.

The position of camera should not be changed as it is fixed (depends on the algorithm).

As when the distance increases the image resolution of detected face decreases.

Face detection and recognition

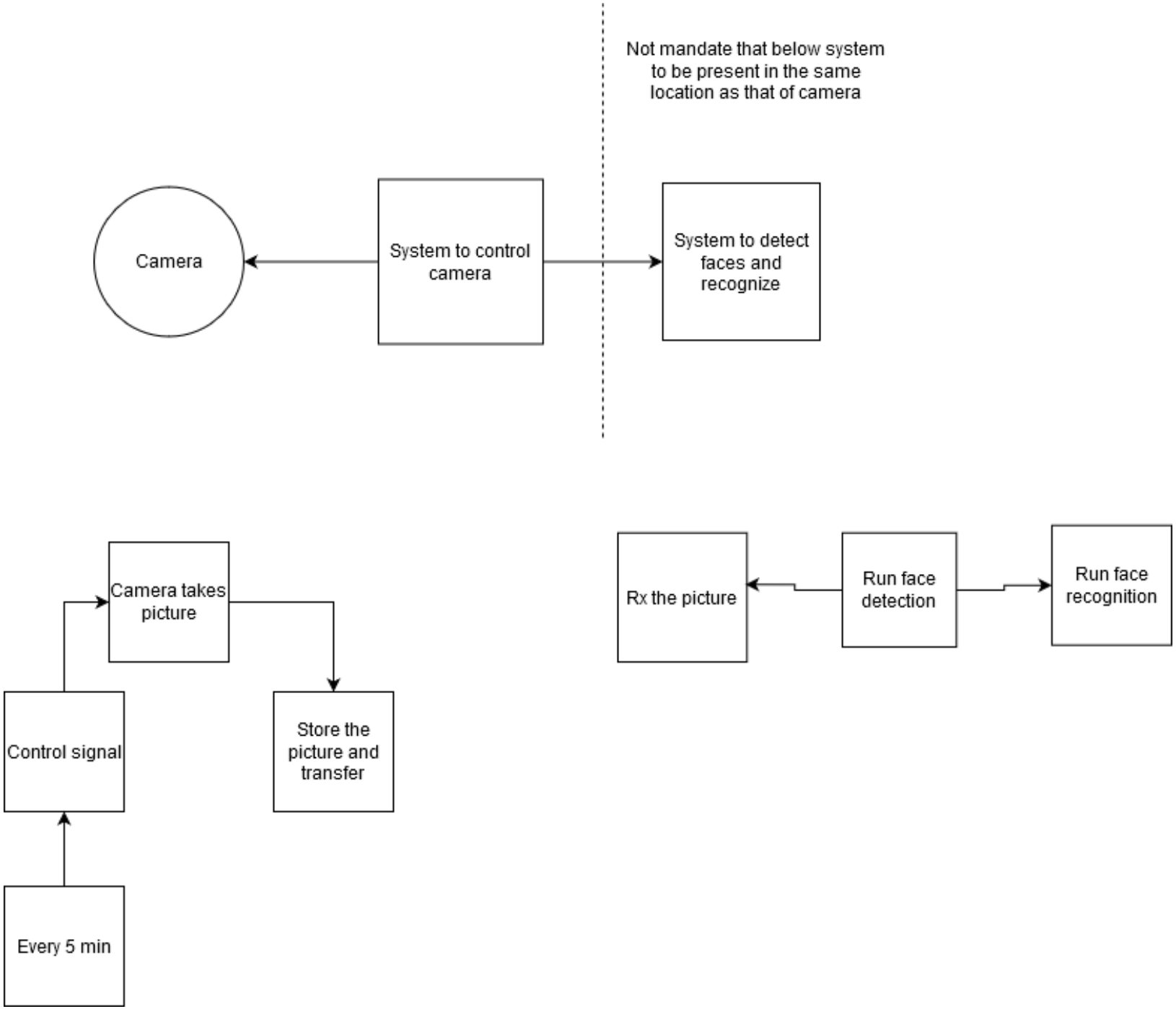
- The face or faces should not be covered.
- There is a maximum distance till which a student can sit for the camera to detect the face beyond that facial recognition won't work.
- When the photo is taken if he/she covers their face then the algorithm can miss the attendance of the person in that time frame.
- If physical characteristics of people change over time — if they gain weight, lose their hair, grow a beard or start wearing glasses, it is tough for face recognition.

Conceptual Design

3 Questions

- What are the needs
  - A reliable system which identifies which student is present in the classroom and gives the attendance based on his presence
- How can the needs be met
  - uses a camera that takes photo of a group of students, and then uses facial recognition for marking attendance for a student
- How well the needs are met
  - found that facial recognition overcomes some of the problems that exist in the previously mentioned bio metric attendance systems subjected to the constraints mentioned in the previous slides

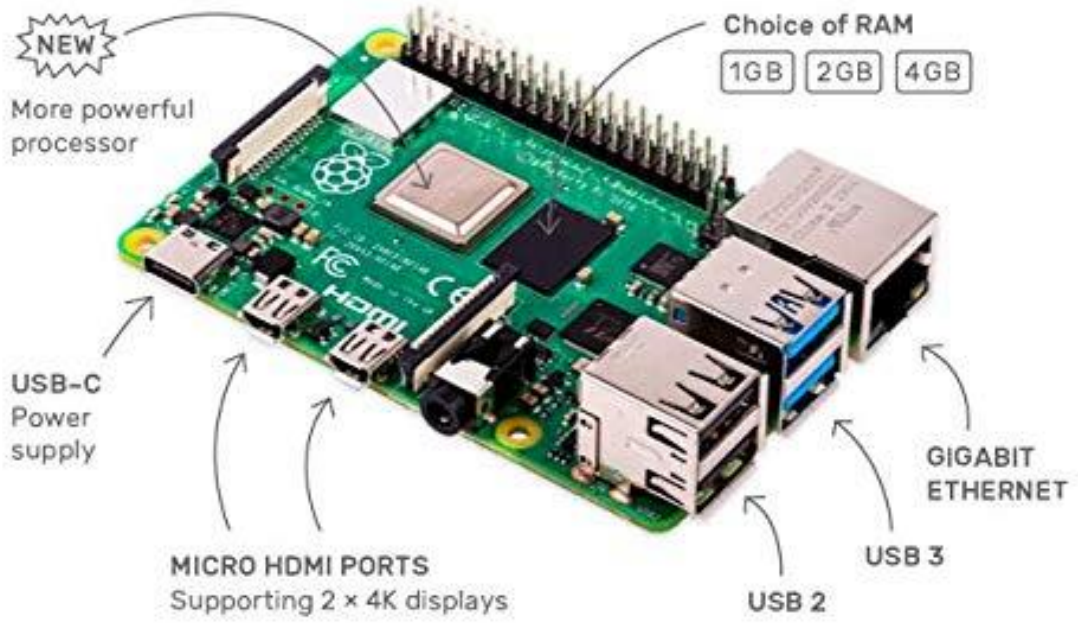
A basic block diagram


Not mandate that below system to be present in the same location as that of camera
Camera
System to control camera
System to detect faces and recognize
Camera takes picture
Store the picture and transfer
Control signal
Every 5 min
Rx the picture
Run face detection
Run face recognition

Our implementation

- Using this 2 algorithms
  - Facial detection - DLIB HOG (Histogram of Oriented Gradients)
  - Facial recognition – Facenet (A deep learning based model that has 128 descriptors)
- Raspberry pi can be programmed to perform the role of all the major sub systems discussed before

Raspberry Pi 8MP Camera Module V2 with Cable



Raspberry pi 4 (4GB RAM)

Estimated Cost

	A	B
1	Components	Price (INR)
2	Raspberry pi 4	5000
3	Raspberry pi 4 camera	1500
4	Fan	500
5		
6	Total	7000
7		

Future work

- In the future we can implement a separate system for running face recognition, connecting the system in class with the other system wirelessly
- Using lidar we can try to improve the facial recognition algorithm which depends on 2D image
- An IR lidar will enable us to detect face even in darkness
- Also we will try to integrate surveillance into this system

References

- C. Ding and D. Tao. Trunk-branch ensemble convolutional neural networks for video-based face recognition. IEEE Trans Pattern Anal Mach Intell, 40(4):1002–1014, apr 2018.
- Y. Chen, Y. Tai, X. Liu, C. Shen, and J. Yang. FSRNet: End-to-end learning face super-resolution with facial priors. In 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2018.
- https://www.learnopencv.com/face-detection-opencv-dlib-and-deep-learning-c-python/
- https://www.electronicscomp.com/raspberry-pi-8mp-noir-camera-module-v2-india?gclid=CjwKCAiA2O39BRBjEiwApB2Ikjhc_K3Lxq-xtrgZNZA4mhJy12pRJj8mdGWV0sX2t-iGZRdeBfWyRBoCfuAQAvD_BwE

Thank You