MULTI-CAMERA BASED ATTENDANCE SYSTEM

IIVP632C Group 15

FACE RECOGNITION ATTENDANCE SYSTEM

MOTIVATION

A STEP TOWARDS
FUTURE

Our aim is to reduce manual processing errors by providing an automated and a reliable attendance system using face detection and recognition model based on multiple cameras

WHY USE VISUAL BASED ATTENDANCE SOFTWARE?

CONVENIENCE

Convenient for teachers as there will be no need to mark attendance manually

EFFICIENCY

Everything is automated and so there will be no proxies or any type of forgery

MONITORING

Monitoring of students will be made easy.

Phases

1. CHECK CAMERA & CAPTURE FACES
2. TRAIN FACES
3. RECOGNIZE FACES & ATTENDANCE
4. AUTOMATIC EMAIL

Phase 1

1

CHECK CAMERA &
CAPTURE FACES

In this phase our model will check for the availability of camera and then capture faces on each camera respectively so that we can collect the face data of the students for which we will be marking the attendance .

Phase 2

2

TRAIN FACES

Here we will train our model on the basis of initially captured face data so that it becomes ready for marking attendance in next phase & such that it will recognise the face of the student in maximum efficient manner possible.

Phase 3

3

Recognise faces and mark attendance

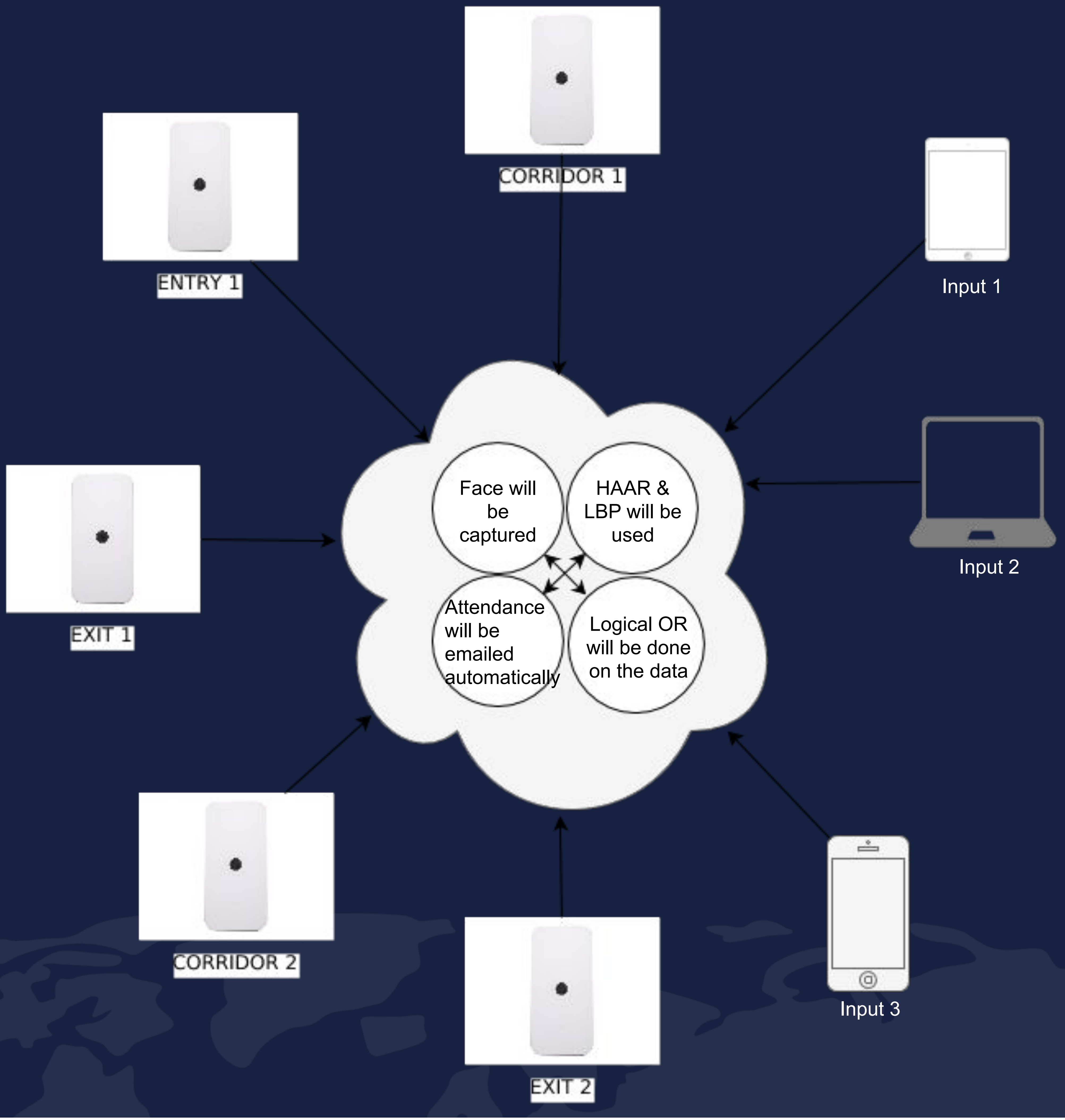
This will be the most important phase of our model. Here all the cameras that have captured the faces will send their data to the server. Now, as we will be having different data from different cameras, The server will implement logical OR operation on the files and if any one of the camera have recognised the faces the server will mark the attendance.

Phase 4

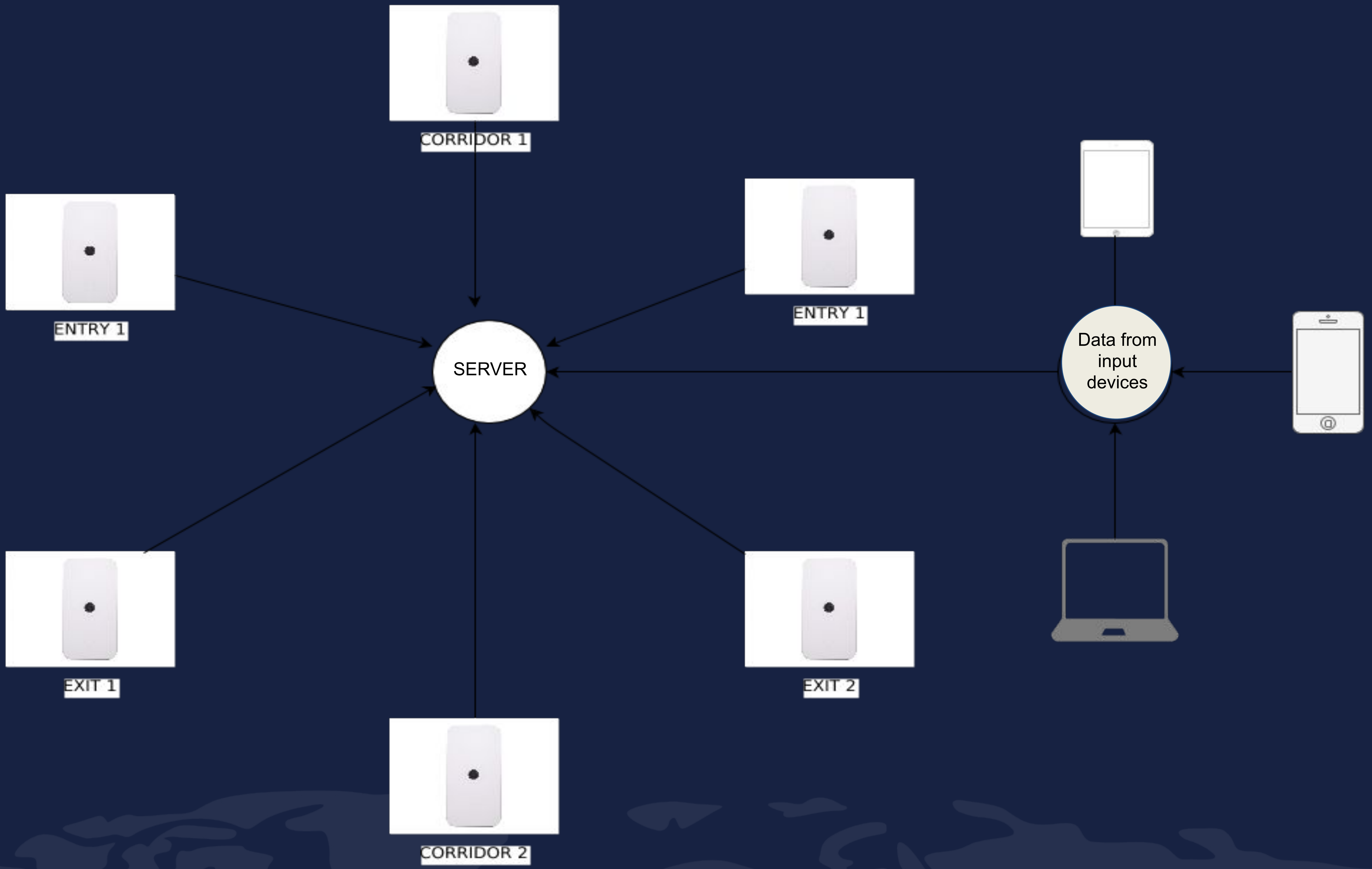
4

Automatic e-mail

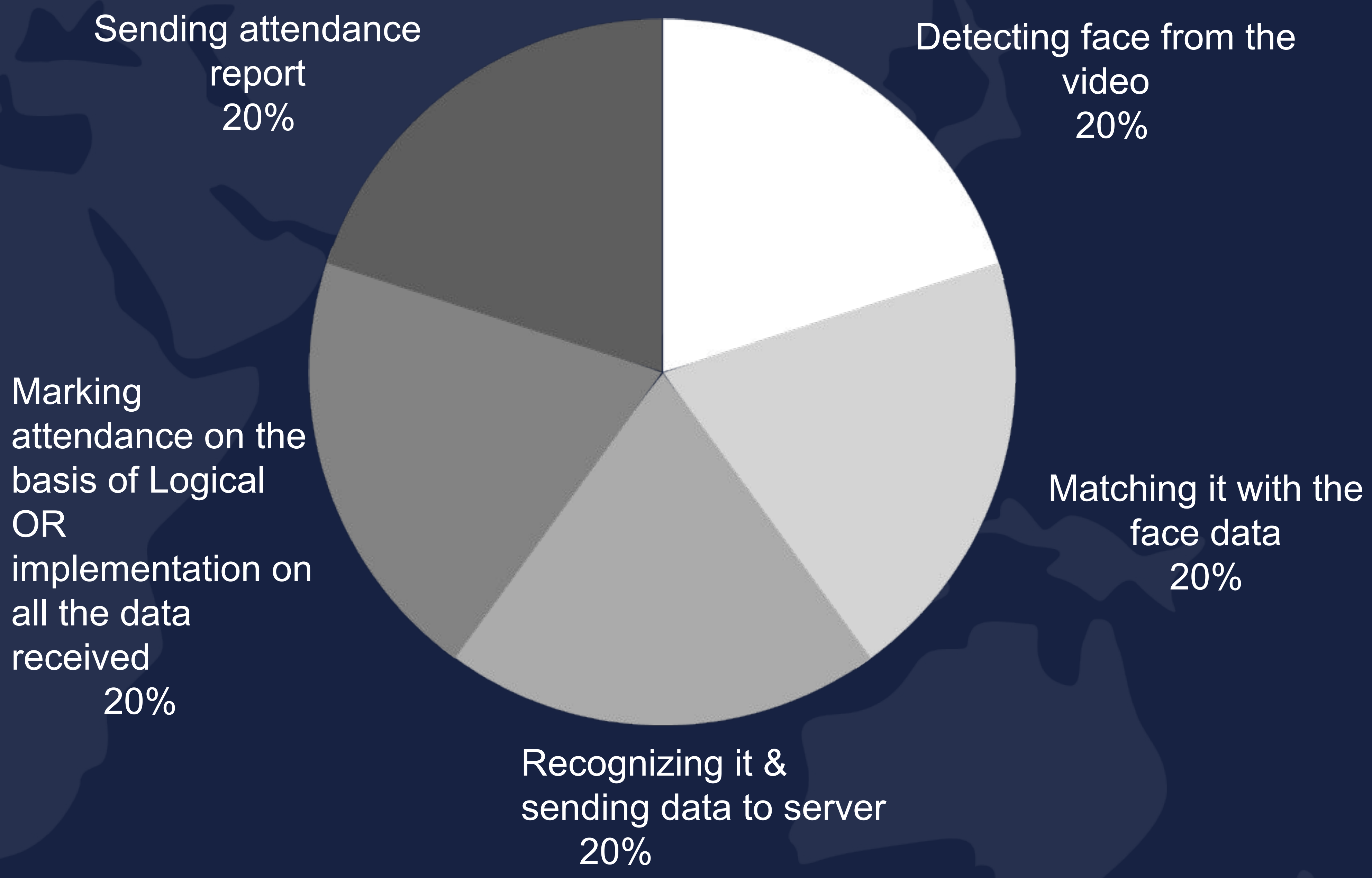
The server will implement logical OR operation to the data it recieved and input the final attendance into the final attendance sheet and will send this as an email to the respective faculty.



CORRIDOR 1
ENTRY 1
Input 1
Input 2
EXIT 1
Face will be captured
HAAR & LBP will be used
Attendance will be emailed automatically
Logical OR will be done on the data
CORRIDOR 2
EXIT 2
Input 3


CORRIDOR 1
ENTRY 1
ENTRY 1
SERVER
Data from input devices
EXIT 1
EXIT 2
CORRIDOR 2

PROJECT DIVISION



we have divided the project in certain stages as shown in pie chart.

TECH USED

BUILD USING

- Python 3.7

SOFTWARE

- pycharm 2019.2
- vsCode

MODULES USED

- OpenCV Contrib4.0.1
- Pillow
- Numpy
- Pandas
- Shutil
- CSV
- yagmail

DATASET USED

We are using the images of enrolled students contained in particular folders mentioned by their names. The model will be trained on these pics and the relevant attendance will be marked.

Face detection & recognition Algorithms used

HAAR CASCADE

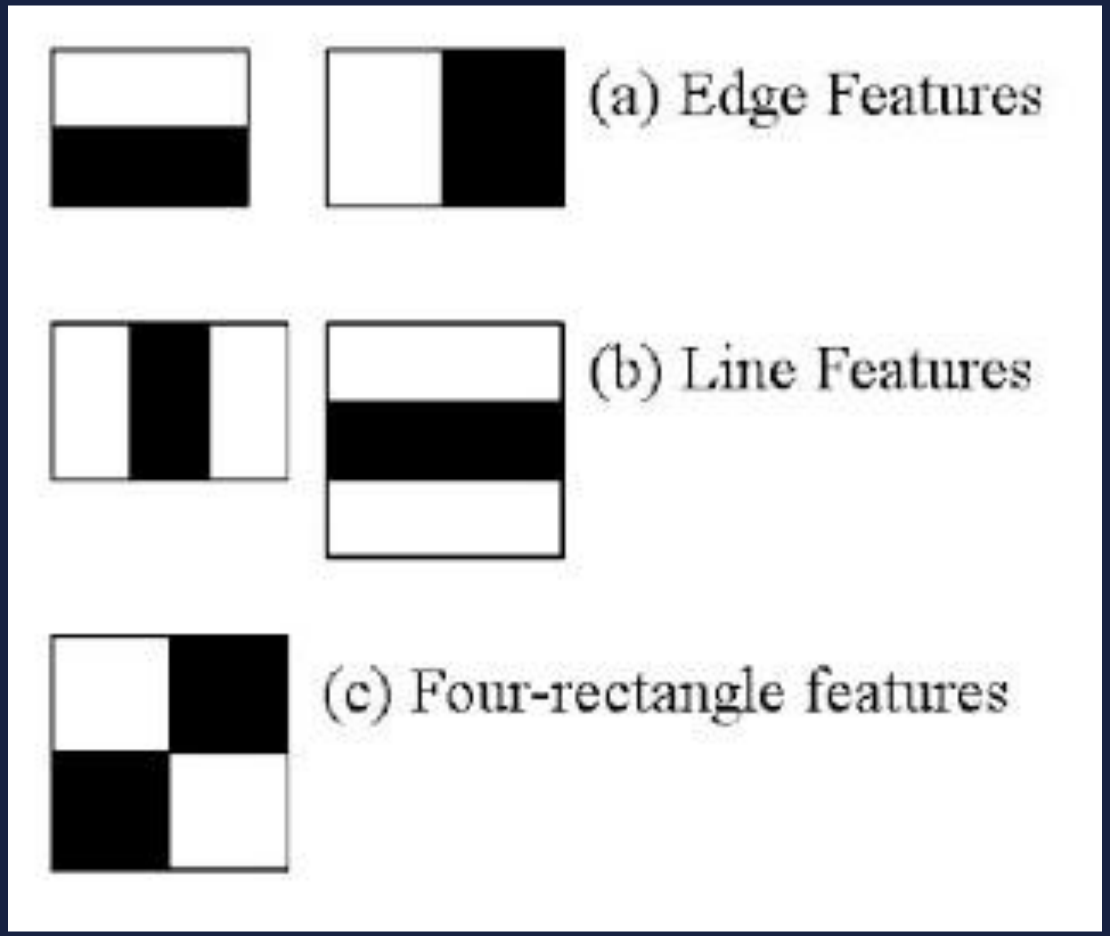
Haar Cascade is a machine learning object detection algorithm used to identify objects in an image or video and based on the concept of features proposed by Paul Viola and Michael Jones in their paper "Rapid Object Detection using a Boosted Cascade of Simple Features"

LBPH (LOCAL BINARY PATTERN HISTOGRAM)

The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameters radius and neighbors

HAAR Cascade

- Basically the HAAR Cascade classifier is an object detection method.
- It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images which is further used to detect objects in other images.
- For face detection, initially the algorithm is feeded with a lot of positive images (images of faces) and negative images (images without faces) to train the classifier, after which, features are extracted from it.
- For the feature extraction, HAAR features are used. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle as shown in the above diagram.
- OpenCV provides a training method or pretrained models, that can be read using the cv::CascadeClassifier::load method.



LBP - Local Binary Patterns

- Local Binary Patterns (LBP) is a visual descriptor used for classification in Computer Vision in Computer Vision.

- A minimal version of LBP feature vector is created in the following manner.
  - Divide the examined window into cells
  - For each pixel in the cell, compare it with each of its neighbours (i.e. 8 neighbours). Follow the neighbours along a circle, i.e., clockwise or counter-clockwise.
  - If the center pixel's value is greater than the neighbour's value, write "0", else "1". This will result in a 8-digit binary number.
  - Compute the histogram, over the cell, of the frequency of each "number" occurring. This histogram can be seen as a 256-dimensional feature vector.
  - Optionally normalize the histogram.
  - Concatenate histogram of all cells. This gives a feature vector for the entire window.

SPECIAL INSTRUCTIONS

STUDENT DETAILS:

There is a file named StudentDetails.csv in a folder named student details The file is not having columns for ID & name. The problem arises when the program run for the first time and automatically create the file StudentDetails.csv. In order to solve this problem just open the file and add the ID & name column manually.

AUTO ATTACHMENT:

This is not considered to be a problem its just before sending the mail automatically we have to change the name. I tried to automate the attachment but results were not found as per required.

References

1. Davis, Michael, et.al (2011). "Real-Time Face Recognition from Surveillance Video." 10.1007/978-3-642-17554-1_8.
2. Hasanen, Sana (2017). “Faces Tracking from Multi-Surveillance Camera” 10.18081/2222-4223/017-12/18-27.
3. N.Sudhakar Reddy, M.V.Sumanth, S.Suresh Babu, "A Counterpart Approach to Attendance and Feedback System using Machine Learning Techniques",Journal of Emerging Technologies and Innovative Research (JETIR), Volume 5, Issue 12, Dec 2018.