**AUTOMATED ATTENDANCE SYSTEM FOR MIT HOSTELS BASED ON FACIAL RECOGNITION**

**PROJECT PROPOSAL**

**BRIEF OUTLINE**

|  |  |
| --- | --- |
| **WHY IS THE PROJECT NECESSARY?** | The current system employed by the MIT hostels is a fingerprint punching system to keep a record of the presence of the hostel immates by the respective block caretakers. This adds on to the responsibilities of the immates to punch everyday on the right time. |
| **OBJECTIVES** | * To completely discard the fingerprint based punching system. * Automated attendance system based on facial recognition. * Deployment of a linear regression model to chisel out the results and prevent the false positives and the true negatives. * Testing of the setup and deployment. |
| **PHASES OF WORK**  **& DESIRED**  **OUTCOME** | * Phase 1: Installation and setup of the NVIDIA Jetson tx1 and the required softwares(OpenCV) with the required number of camera modules(3 in number for the initial stage) * Phase 2: Determination of best fit position of the cameras in and around the hostel lobby. * Phase 3: Training the ML model with available pictures of the immates. * Phase 4: A week long training of the ML model at the hostel lobby. * Phase 5: Final deployment. |

**INTRODUCTION**

The Task of Face Recognition is done by C++ Program written using OpenCV library.

The stages of this project include:

* A C++ program to detect and store faces. *(Detection)*
* A Python Script to maintain and link available faces. *(Linking)*
* A C++ program to fetch faces from a camera and compare them with available database. *(Recognition)*
* A Python Script to update the record on Google Spreadsheets over a secure wireless connection. *(Uploading)*

The primary processing unit that will be used in the project is the **Nvidia Jetson TX1**

Sees Recognises, Keeps a Remembers record

**Sees**:

The video of the lobby is captured through a webcam connected to the Nvidia Jetson tx1.

An OpenCV program running on the device will be recognising the faces in each frame of the video.

Algorithm employed for facial recognition:

**Haar’s Frontal Face classifier**

The task includes:

* Calculating the number of faces
* Segmenting, cropping and resizing the images

**Remembers:**

The program creates a database only when a face is detected. Each database will be primary to a single student and will contain the images of only that student.

The database structure will be as follows:

|-- student1

| |-- 1.img

| |-- ...

| |-- 29.img

|-- student2

| |-- 1.img

| |-- ...

| |-- 29.img

...

|-- student40

| |-- 1.img

| |-- ...

| |-- 29.img

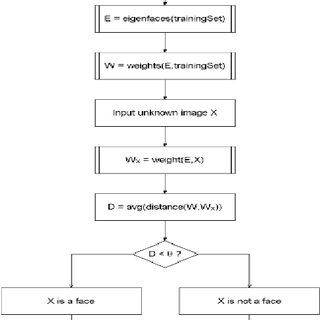
After extraction of the facial image, the image is reduced to a size of 300px x 300 px and is greyscaled.

**Recognition:**

The program fetches live feed from the default imaging device and processes it frame by frame.  
  
The first task that the program performs is to train its Two classifiers on the training database and labels of images.The Two algorithms used are:

* Eigenface Algorithm   ( Principal Component Analysis)
* Fisher Face Algorithm( Linear Discriminant Analysis )

The Eigenface Algorithm is used to find the similarities between multiple images of an individual.

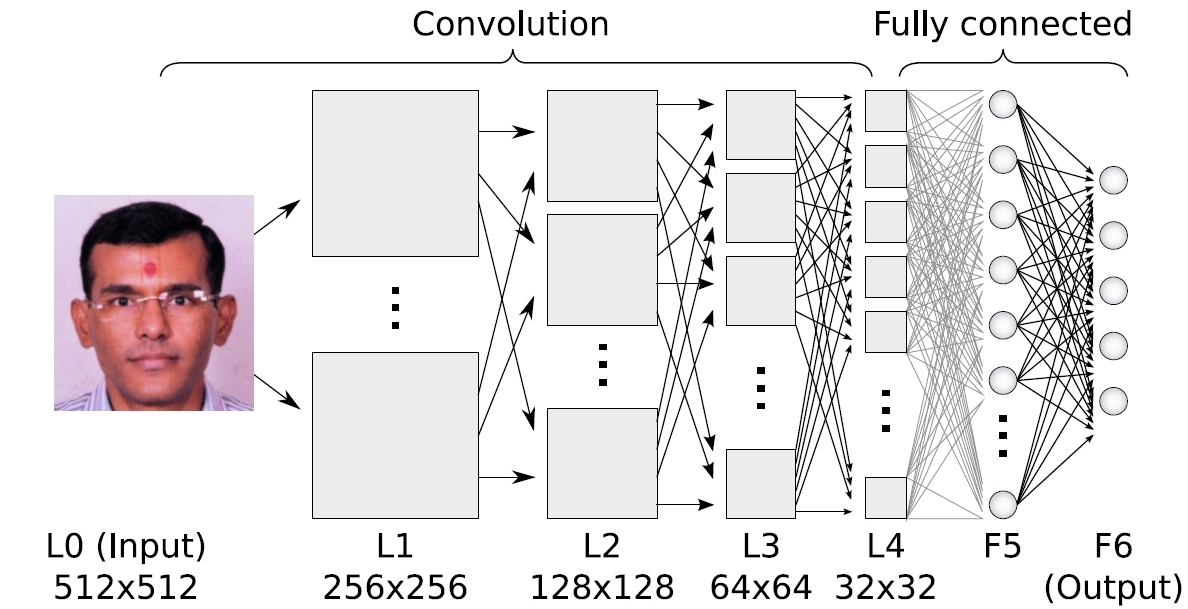


The Fisher Face Algorithm is used to find the difference between those same multiple images of that individual.

The collective results of these two algorithms is what is used in determining the individual.

The Haar’s cascade method is run to segment the facial images and then they are evaluated by both the algorithms.

The discriminating edges of a face that the haar classifier uses.

Representation of Facial Recognition Using Deep Convulation Neural Network

**Storage of data in Google Drive:**

A python script will be used to connect to the google cloud and update the attendance spreadsheet.

Packages to be used:

* Oauth2client  (Google Cloud Authentication Client)
* Gspread   (Google Spreadsheet API client)
* PyOpenSSL (Python Open SSL package)

The script creates a command line argument which includes the predicted result(student unique id) and the time at which the frame was captured.

Few more problems to be solved by the script is:

* Date Row not found -> Create row for Current Date. *(When taking attendance on a new day)*
* Roll No not found -> Create column for Roll No. *(When database is updated)*
* Date Row found, Roll No column not found -> Add Roll No column and write "Present" in current date row   *(Database updated during current day)*
* Date Row not found, Roll No. Column Found -> Add Date Row and write "Present" in current Roll No. column  *(Database Intact, Day changed )*

**Conclusion:**

This automated attendance system based on facial recognition can be utilised anywhere, which keeps a track of a large number of people. This helps in reducing effort required and also time spent. It can also be deployed in the MIT Lecture halls where the attendance of the student gets updated automatically in the SLcM portal, preventing any margin for error. Since it deploys a ML model, the results and the recognition time will get more and more refined with time and data.

**Referrence:**

<https://ijcsmc.com/docs/papers/February2014/V3I2201468.pdf>

<https://www.researchgate.net/publication/241608617_Face_Recognition-based_Lecture_Attendance_System/download> (research paper)

https://www.irjet.net/archives/V5/i4/IRJET-V5I41022.pdf