D.K.T.E. Society's Textile and Engineering Institute, Ichalkaranji.

(An Autonomous Institute)

Department of Information Technology

2019-20

Project Report On

Smart Security System

Under the Guidance Of

Prof. A. S. Shelar

Submitted By:

1. Vinay Sakhare	16UIT12046XX
2. Shivani Patil	16UIT11041XX
3. Rutuja Prabhudeasi	16UIT11044XX
4. Akshaykumar Patil	16UIT12037XX
5. Kanishk Shedsale	16UIT12049XX

Prof. A.S.Shelar	Prof.(Dr).D.V.Kodavade	Prof.(Dr).P.V.Kadole
(Project Guide)	**(HOD)**	**(Director)**

DKTE

Promoting Excellence in
Teaching, Learning & Research

YEAR 2019-2020

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project report entitle “Smart Security System“ is record of project work carried out in this college by,

Mr. Vinay Sakhare	(16UIT12046XX),
Miss. Shivani Patil	(16UIT11041XX),
Miss. Rutuja Prabhudesai	(16UIT11044XX) ,
Mr. Akshaykumar Patil	(16UIT12037XX) ,
Mr. Kanishk Shedsale	(16UIT12049XX)

In the partial fulfilment of the requirement for degree of BACHELOR OF ENGINEERING in INFORMATION TECHNOLOGY of SHIVAJI UNIVERSITY, KOLHAPUR. This project report is record of their own work carried out under my supervision and guidance during academic year 2017-2018.

Prof. A.S. Shelar
[Project Guide]

Prof. (Dr.) D. V. Kodavade
[Head of the Department]

Prof. (Dr.) P.V. KADOLE
[Director]

DECLARATION

We hereby declare that, the project work report entitled “Smart Security System” which is being submitted to D.K.T.E. Society’s Textile and Engineering Institute Ichalkaranji. An Autonomous Institute affiliated to Shivaji University, Kolhapur is in partial fulfilment of degree B.Tech (I.T.). It is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for the award of any degree. Further, we declare that we have not violated any of the provisions under Copyright and Piracy/Cyber/IPR Act amended from time to time.

Name	Roll No	Signature
1. Vinay Sakhare	16UIT12046XX	
2. Shivani Patil	16UIT11041XX	
3. Rutuja Prabhudeas	16UIT11044XX	
4. Akshaykumar Patil	16UIT12037XX	
5. Kanishk Shedsale	16UIT12049XX	

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to our project coordinator PROF.A.S.SHELAR for his guidance, invaluable support and encouragement throughout the project. We would also like to thank our Head of the Department for providing us this opportunity to work on a project.

This project would have been impossible without our project guide and we want to extend sincere thanks to her for her guidance and constant supervision as well as for providing necessary information regarding the project.

We would also like to express our gratitude towards our parents and our college for their kind co-operation and encouragement which helped us in completion of this project. Our thanks and appreciations also go to our colleague in developing the project and people who have willingly helped us out with their abilities.

Thank you,

Mr. Vinay Sakhare	(16UIT12046XX),
Miss. Shivani Patil	(16UIT11041XX),
Miss. Rutuja Prabhudesai	(16UIT11044XX) ,
Mr. Akshaykumar Patil	(16UIT12037XX) ,
Mr. Kanishk Shedsale	(16UIT12049XX)

ABSTRACT

This project gives an outline for an automatic system to control and secure specific area, based on digital image processing with the help of Internet of Things and AI. The system consists of a sensor, digital camera, database and the smart phone. Sensors are placed in the specific area which alerts camera to start recording the activity in that area. Then video is than divided into frames. Image analysis is performed to detect and recognize and match the image with the stored dataset of the authenticated people. If the image captured does not match with the dataset then an alert message is send to the owner of that area.

INDEX

INTRODUCTION

In recent years, the security constitutes the most important section of our lives. Automation of a home is an exciting field for security applications. This area has developed with new technologies like Internet of things (IoT) and Artificial Intelligence. In IoT, each device behaves as a small part of an internet node and each node communicate and interact. Artificial intelligence (AI) is the simulation of human intelligence processes by machines, especially computer systems. These processes include learning (the acquisition of information and rules for using the information), reasoning (using rules to reach approximate or definite conclusions) and self-correction.

Currently, security cameras are used in order to construct safety areas, cities, and homes. The camera records the series of images and, when a motion is detected. An IoT-based system is combined with computer vision in order to detect the people. A Raspberry PI 3 card with the size of a credit card was used for this purpose. A motion is detected by the PIR sensor mounted on the Raspberry PI. PIR sensor helps to monitor and get alerts when movement is detected. A human face and body detector is first proposed, based on a simple probabilistic model, to approximately estimate human face and body regions. Afterward, human is detected in the captured image and compared with the stored dataset whether the target human is authenticated or not. An efficient algorithm for humans' retrieval from large video databases is presented.

PROBLEM STATEMENT

To provide smart security system to the specific area using IOT (sensors) and AI (human detection).

OBJECTIVE

- To sense the motion in the specific area.
- To capture the image and recognize the person.
- To generate an alert for unknown people.

PROBLEM DESCRIPTION

The sensors will sense the motion in the specified area and activate the cameras. These cameras will record the activity and send it for further image analysis. If the human detected from that image does not match with the authenticated people then it generates an alert message as well as alert alarm.

TIMELINE OF THE PROJECT

TOPIC	START DATE	END DATE
Domain selection	06/08/2019	13/08/2019
Domain finalization	06/08/2019	13/08/2019
Selection of problem statement	13/08/2019	20/08/2019
Finalization of problem statement	20/08/2019	03/09/2019
Study of Research Paper	03/09/2019	07/09/2019
Documentation of synopsis	07/09/2019	14/09/2019
Requirement analysis	14/09/2019	17/09/2019
System requirement	17/09/2019	21/09/2019
Module identification	21/09/2019	24/09/2019
Architecture module	24/09/2019	29/09/2019
Implementation (25%)	29/09/2019	12/10/2019
Testing (25%)	12/10/2019	15/10/2019
Implementation (50%)	07/01/2020	28/01/2020
Testing (50%)	28/01/2020	31/01/2020
Implementation (75%)	31/01/2020	14/02/2020
Testing (75%)	14/02/2020	22/02/2020
Implementation (100%)	22/02/2020	02/03/2020
Testing (100%)	02/03/2020	10/03/2020
Report making	10/03/2020	21/03/2020

COST OF THE PROJECT

Sr No.	**Equipment**	**Details**	Price(Rs)
1.	Electricity	15(kWh) * 10(days)	150kWh/month
2.	Internet	500 (per month) ***3**	1500
3.	Hardware	3000(Raspberry pi) + 1000(sensors , camera)	4000

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy. Once these things are satisfied, then next steps are to determine which operating system and language can be used to developing the tools. Once the programmer start building the tool the programmer need lots of external support. This support can be obtained from senior friends, teachers, from book or from websites. Before building the system, the above consideration are taken into account for developing the proposed system.

We can see the papers as maintained below.

This project adds mainly four features: security, safety, control and monitoring to home automation. The security system, proposed, is low cost, low power consuming system. This system can easily provide high level of security as it combines two modern technologies together i.e. Face recognition and IoT.

In this paper we develop this system to giving access into a home for authenticated users. The training data are initially collected from social networks.

The accuracy of the classifier is incrementally improved as the user starts using the system. By using a deep learning framework - Tensor Flow, it becomes easy to reuse the framework to adopt with many devices and applications

In [3], instead of providing raw pixel values as input, only the extracted facial features are provided. The proposed method is tested on Yaleface database which consists of black and white images of 15samples, each having 11 images in different expressions making a total of 165 images

In [4] used Haar-like features for face detection and Local Binary Pattern Histogram (LBPH) for face recognition. The system also includes a web- based remote accessing, an authentication module, and a bare-bones embedded IoT(Internet of Things) server. Presence of the owner at all times is not possible, and where a remote authentication and control is desired.

REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENTS

Functional requirements are features that the system will need in order to deliver or operate. In the case of this project, it was important to gather some requirements that will be needed to achieve the objectives set out previously. With client (user) story a use case analysis was implemented which resulted in the following functional and non-functional requirements were captured. The functional requirements have been gathered from the user story developed from the minutes collected during meetings with the client and are outlined here.

- Capture face images via webcam or external USB camera.
- A professional Camera
- Faces in the image must be detected.
- The faces must be detected in bounding boxes.
- Store the faces to a folder.
- Load faces on database.
- Train faces for recognition.
- Perform recognition for faces stored on database.
- Compute recognition rate of the system.
- Generate alarm in the case of unknown person.
- Send notification on the owner's mobile phone.

SOFTWARE AND HARDWARE REQUIREMENTS

Hardware requirement:

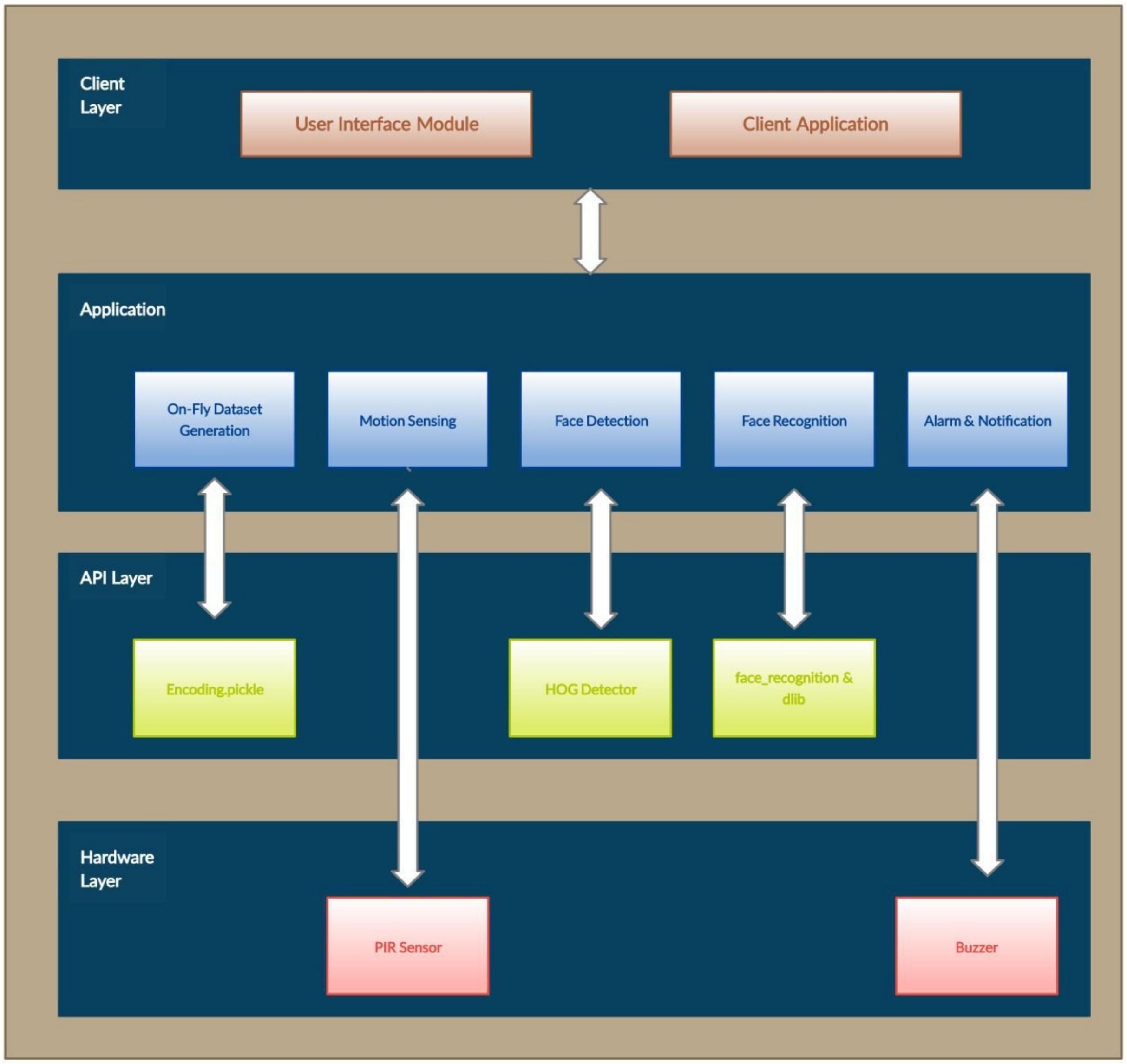
- Laptop with high Ram.
- Camera with good quality
- Raspberry pi
- PIR Sensors
- Buzzer

Software requirements

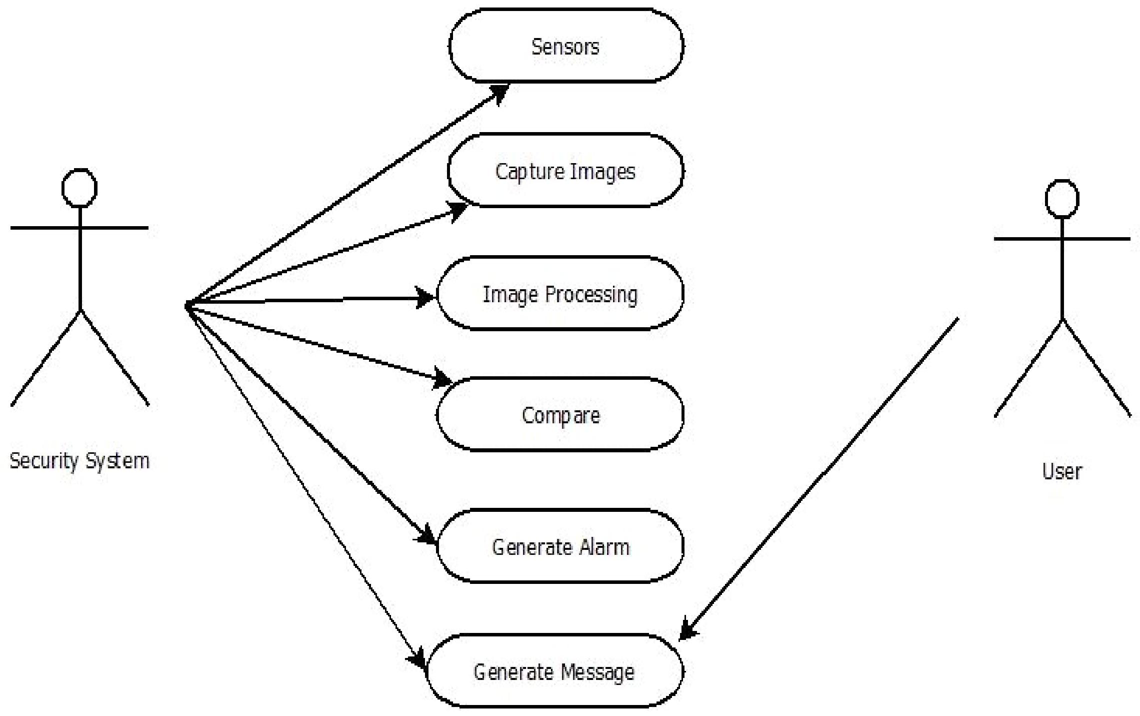
- Microsoft Windows 7 / 8 / 10.
- Python Idle

SYSTEM DESIGN

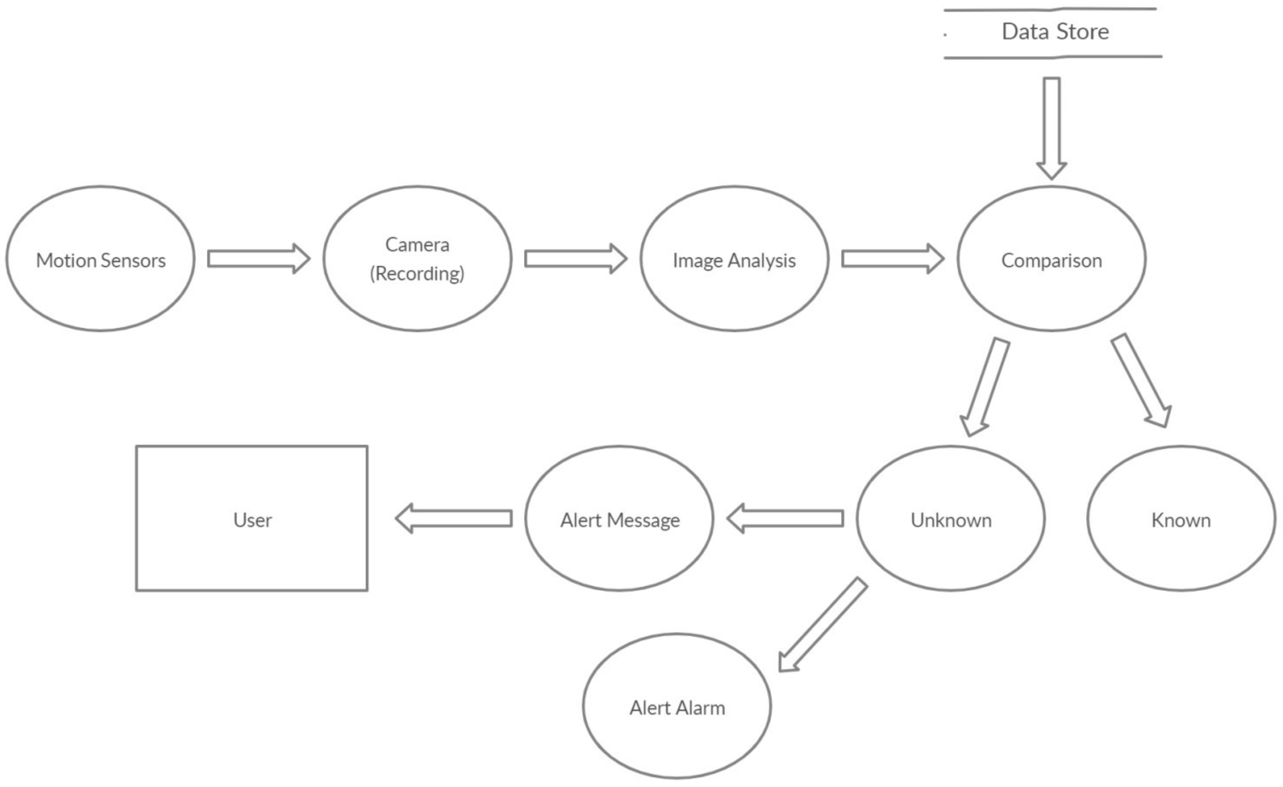
SYSTEM ARCHITECTURE



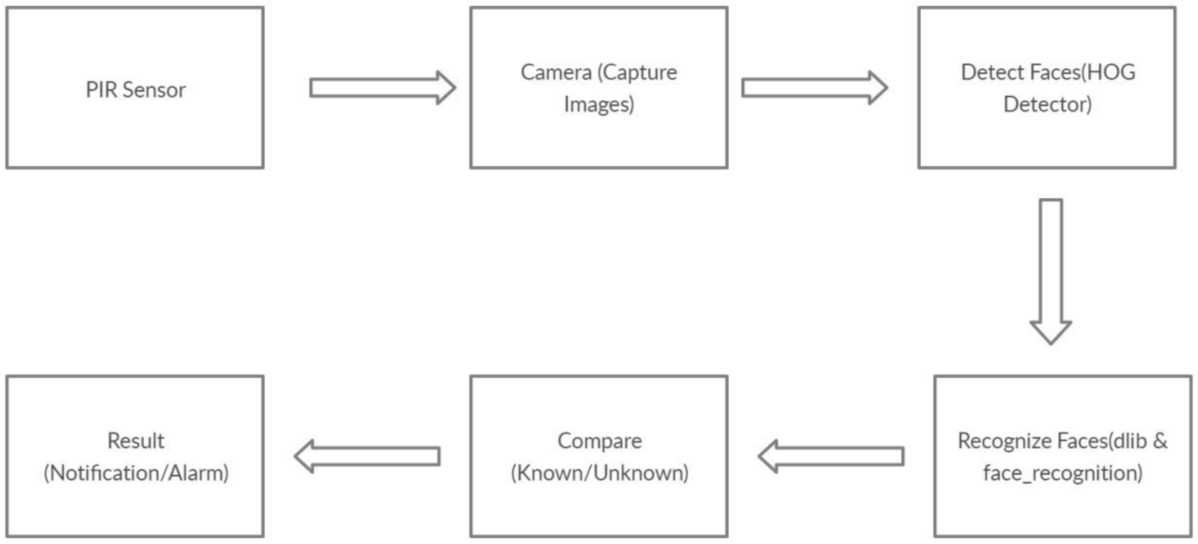
USE CASE DIAGRAM



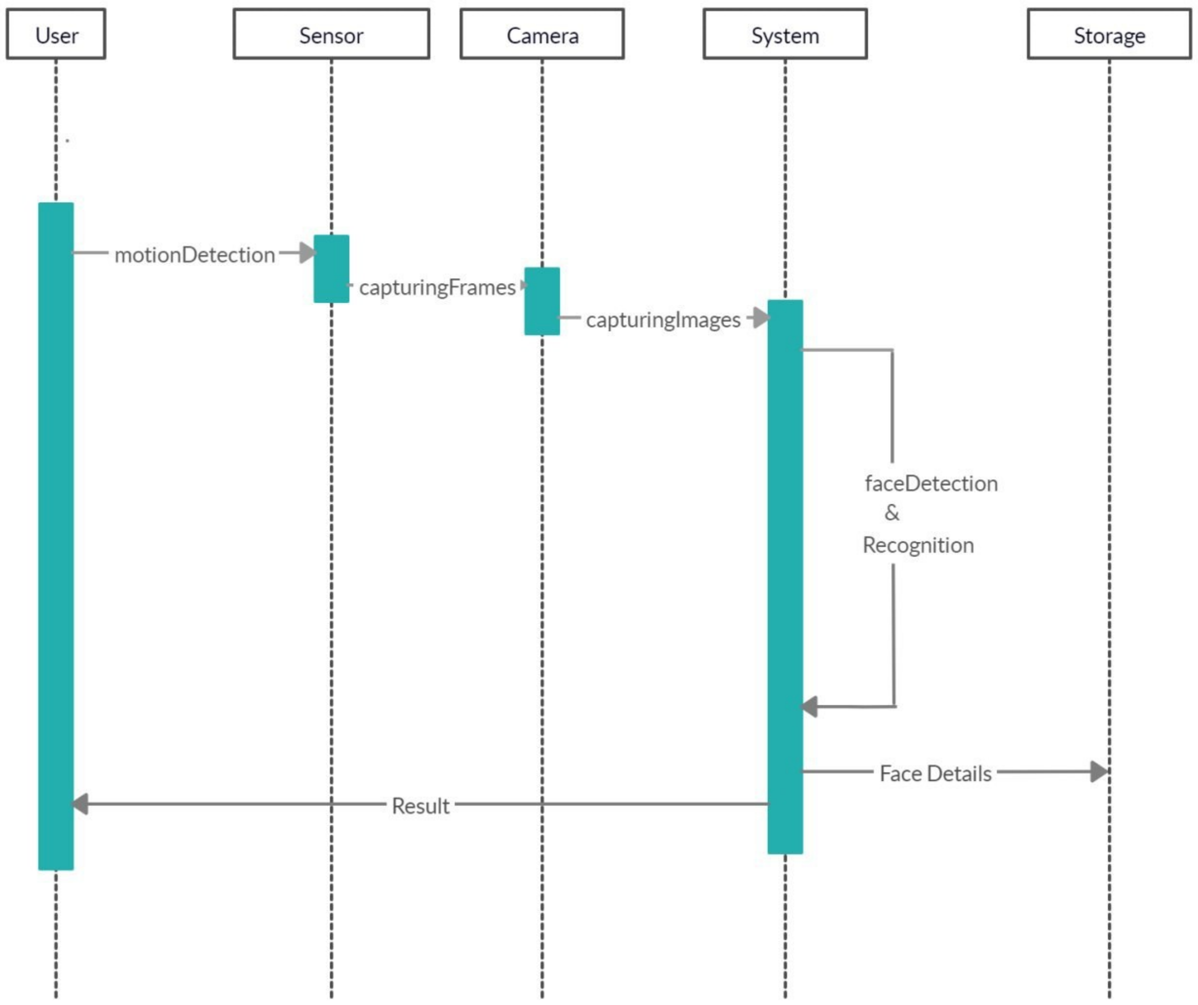
DATA FLOW DIAGRAM



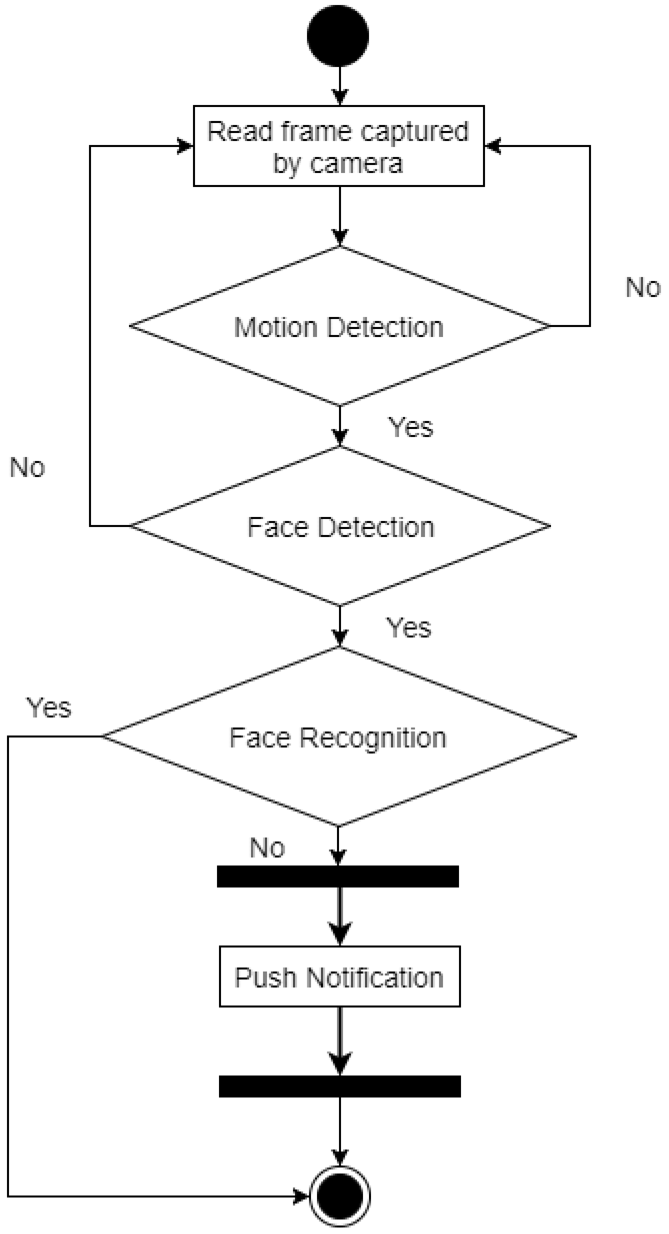
DEPLOYMENT DIAGRAM


PIR Sensor
Camera (Capture Images)
Detect Faces(HOG Detector)
Recognize Faces(dlib & face_recognition)
Compare (Known/Unknown)
Result (Notification/Alarm)

SEQUENCE DIAGRAM



ACITIVITY DIAGRAM



IMPLEMENTATION DETAILS

ALGORITHM DESCRPTION

HOG Lately, the usage of side direction histograms in the concept of image classification is getting popular. Likewise, as the first time the usage of intensive and local gradient direction histograms (HOG) is proposed by Shashua and Dalal. At this method the purpose is to recognize the image as the group of local histograms [1, 9]. These histograms engage the numbers at the area of image's local gradient directions that are head out. The realization of HOG is in order of: x The reception of gradient from image x The set off histogram directions for specified locations x The normalization of histograms in the specified in location groups Moreover, to normalize the local histograms is made at the area that is named block at inside image for progressing the performance by measuring the intensity in this block area.

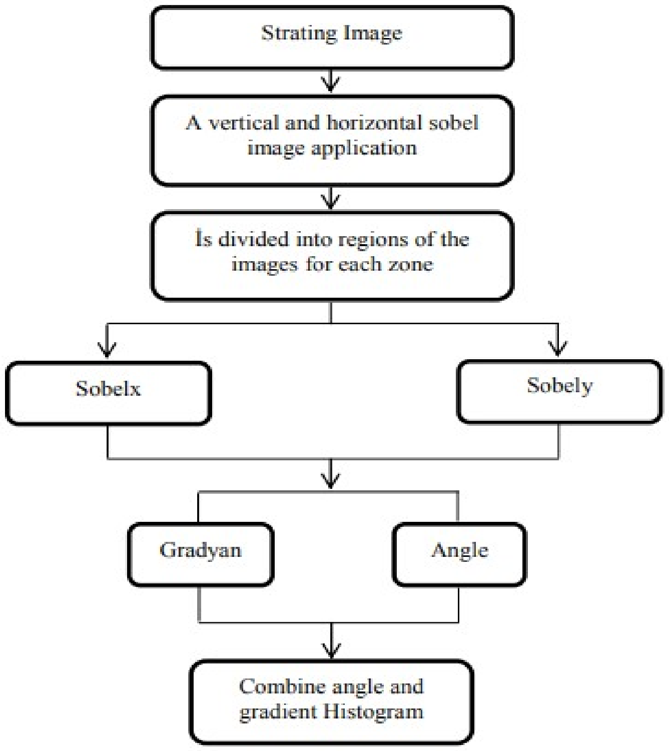

Strating Image
A vertical and horizontal sobel image application
İs divided into regions of the images for each zone
Sobelx
Sobely
Gradyan
Angle
Combine angle and gradient Histogram

Figure.1: HOG algorithm flowchart [9].

HOG is a type of “quality descriptor”. The target of a quality descriptor is to generalize the object in that way that the same object produces as close as possible to the same characteristic descriptor when viewed under different circumstances. This makes the categorization task easier. The HOG descriptor was illustrated by Dalal & Triggs in 2005 as a characteristic set for object identification tasks. At that time, this narrative descriptor outperformed existing characteristic sets for human recognition significantly. The main idea is that local object manifestation and form is characterized by the allocation of local intensity gradients or border directions, without precise knowledge of the equivalent gradient or edge positions [1]. HOG is one of the recognized characteristics for object recognition. HOG characteristics are calculated by taking direction histograms of edge intensity in a local region [2]. Local object appearance and form can often be characterized rather well by the allocation of local intensity gradients or edge detection. HOG characteristics are calculated by taking direction histograms of edge intensity in local region. HOG characteristics are used in the SIFT descriptor suggested by Lowe [6]. Mikolajczyk et al. stated in [7] that the best corresponding results were found by the SIFT descriptor

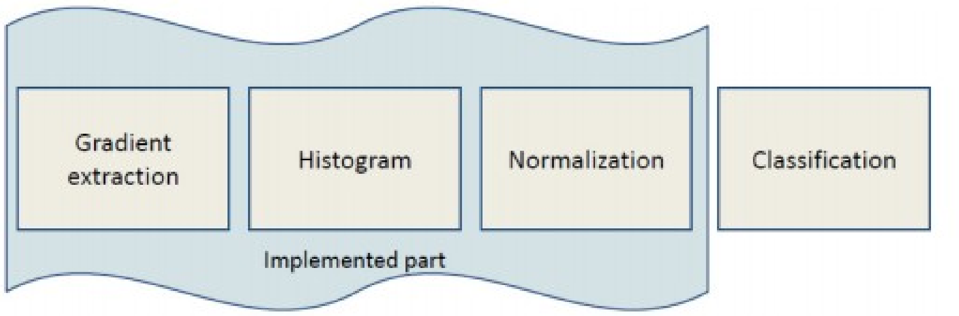


Figure.2: Histograms of Oriented Gradients for Human Detection [2].

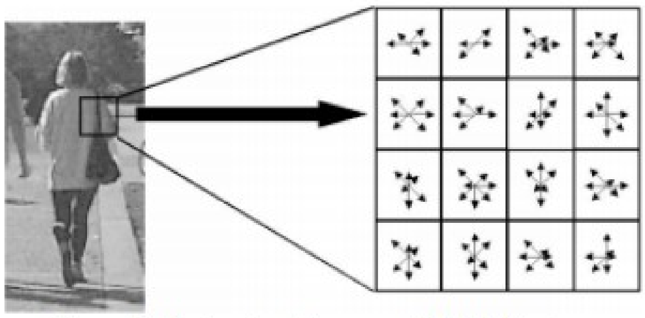


Figure.3. Extraction Process of HOG features

.

The HOG characteristics are taken out from local regions with 16 × 16 pixels. Histograms of edge gradients with 8 preferences are calculated from each of 4 × 4 local cells. The edge gradients and directions are attained by utilizing Sobel filters. Thus, the total number of HOG characteristics becomes 128 = 8 × (4×4).

FACE RECOGNIZATION

Every Machine Learning algorithm takes a dataset as input and learns from this data. The algorithm goes through the data and identifies patterns in the data. For instance, suppose we wish to identify whose face is present in a given image, there are multiple things we can look at as a pattern:

- Height/width of the face.
- Height and width may not be reliable since the image could be rescaled to a smaller face. However, even after rescaling, what remains unchanged are the ratios – the ratio of height of the face to the width of the face won't change.
- Colour of the face.
- Width of other parts of the face like lips, nose, etc.

Clearly, there is a pattern here – different faces have different dimensions like the ones above. Similar faces have similar dimensions. The challenging part is to convert a particular face into numbers – Machine Learning algorithms only understand numbers. This numerical representation of a "face" (or an element in the training set) is termed as a *feature vector*. A feature vector comprises of various numbers in a specific order.

As a simple example, we can map a "face" into a feature vector which can comprise various features like:

- Height of face (cm)
- Width of face (cm)
- Average color of face (R, G, B)
- Width of lips (cm)
- Height of nose (cm)

Essentially, given an image, we can map out various features and convert it into a feature vector like:

Height of face (cm)	Width of face (cm)	Average color of face (RGB)	Width of lips (cm)	Height of nose (cm)
23.1	15.8	(255, 224, 189)	5.2	4.4

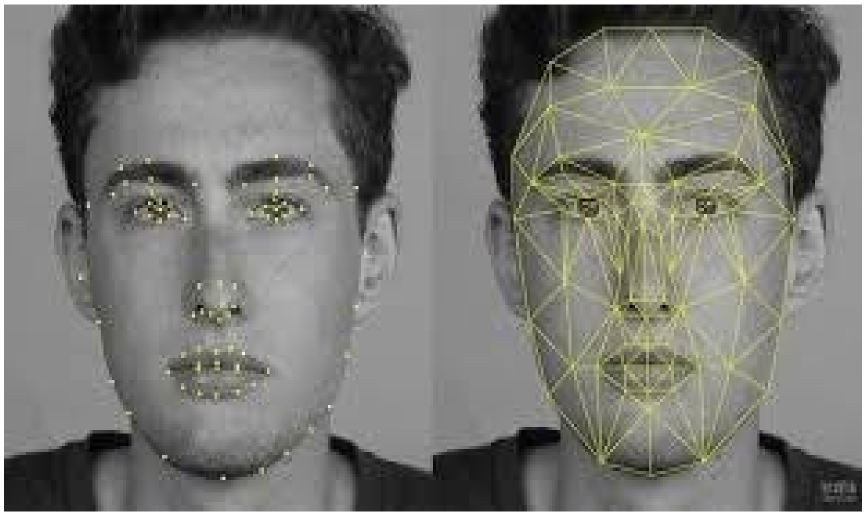
So, our image is now a vector that could be represented as (23.1, 15.8, 255, 224, 189, 5.2, 4.4). Of course there could be countless other features that could be derived from the image (for instance, hair color, facial hair, spectacles, etc). However, for the example, let us consider just these 5 simple features.

Now, once we have encoded each image into a feature vector, the problem becomes much simpler. Clearly, when we have 2 faces (images) that represent the same person, the feature vectors derived will be quite similar. Put it the other way, the “distance” between the 2 feature vectors will be quite small.

Machine Learning can help us here with 2 things:

1. *Deriving the feature vector*: it is difficult to manually list down all of the features because there are just so many. A Machine Learning algorithm can intelligently label out many of such features. For instance, a complex features could be: ratio of height of nose and width of forehead. Now it will be quite difficult for a human to list down all such “second order” features.
2. *Matching algorithms*: Once the feature vectors have been obtained, a Machine Learning algorithm needs to match a new image with the set of feature vectors present in the corpus.

Now that we have a basic understanding of how Face Recognition works, let us build our own Face Recognition algorithm using some of the well-known Python libraries.

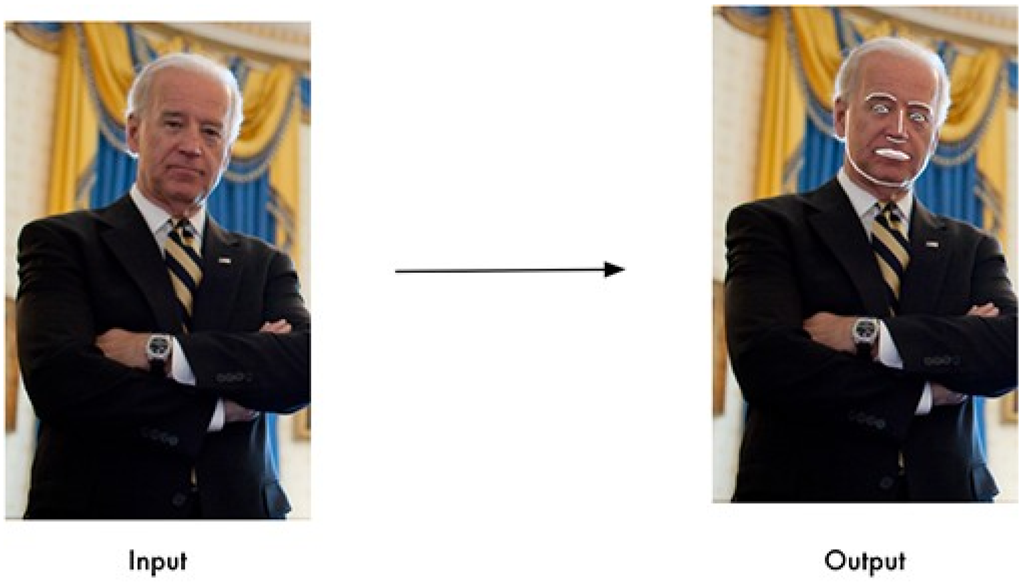


face_recognition library in Python can perform a large number of tasks:

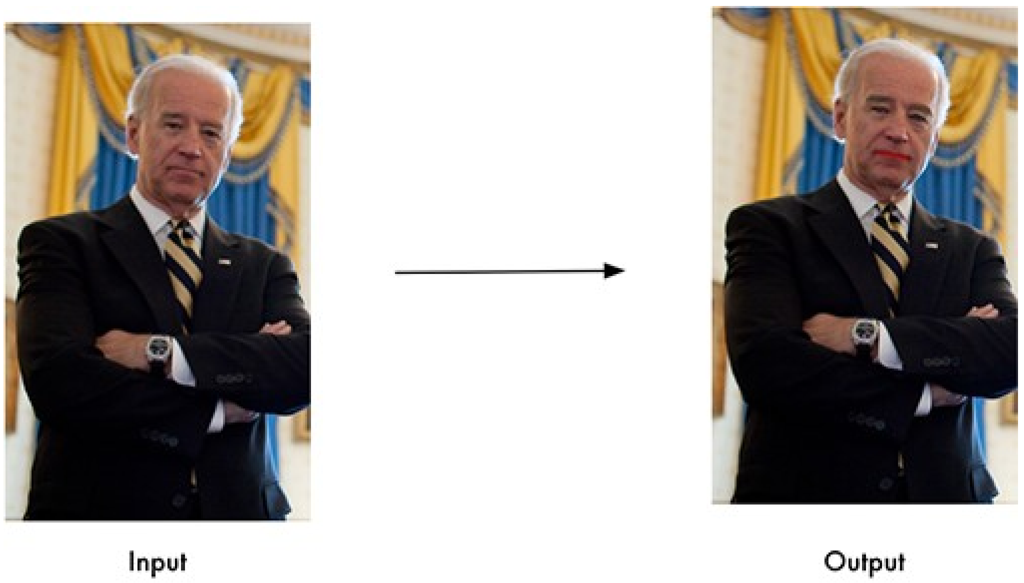
- Find all the faces in a given image
- Find and manipulate facial features in an image
- Identify faces in images
- Real-time face recognition

Find and manipulate facial features in pictures

Get the locations and outlines of each person’s eyes, nose, mouth and chin.



Finding facial features is super useful for lots of important stuff. But you can also use for really stupid stuff like applying digital make-up (think 'Meitu'):



Identify faces in pictures

Recognize who appears in each photo.

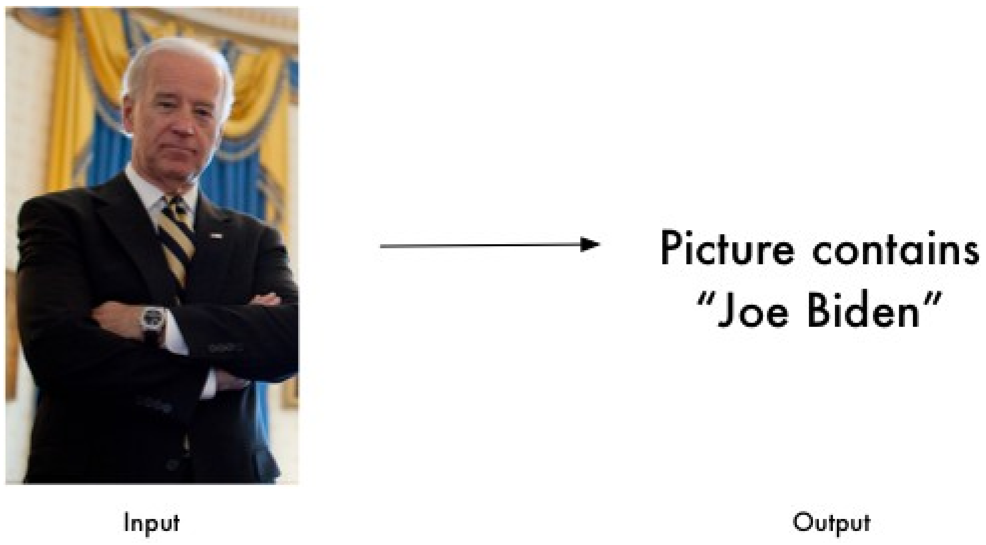
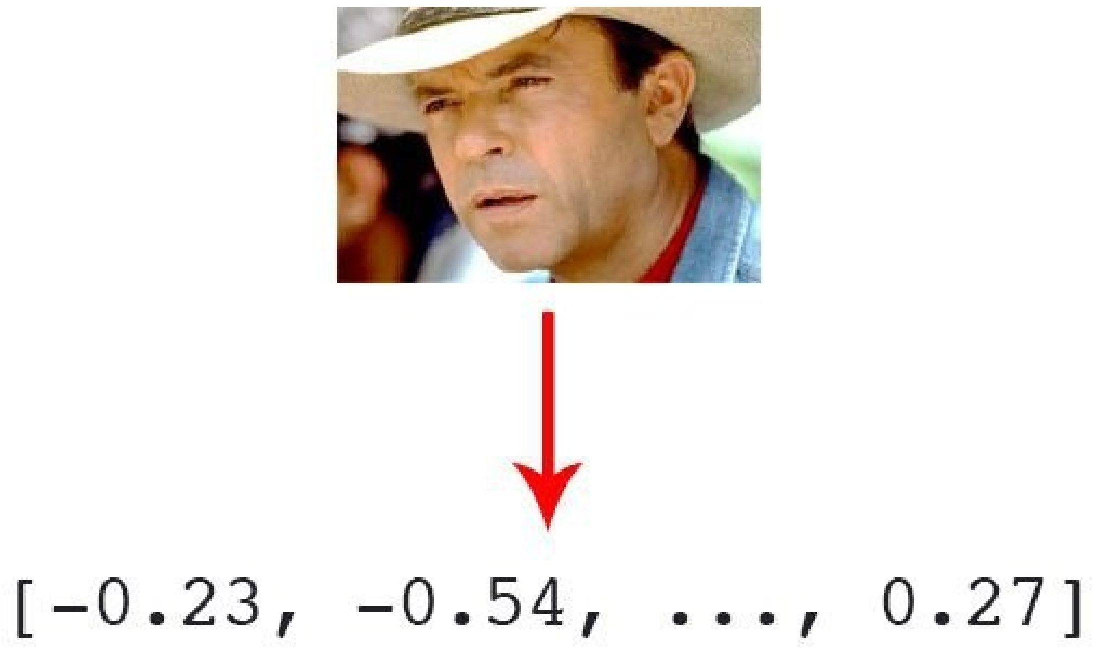


Figure: Facial recognition via deep learning and Python using the face_recognition module method generates a 128-d real-valued number feature vector per face.



Before we can recognize faces in images and videos, we first need to quantify the faces in our training set. Keep in mind that we are not actually training a network here — the network has already been trained to create 128-d embedding on a dataset of ~3 million images.

We certainly could train a network from scratch or even fine-tune the weights of an existing model but that is more than likely overkill for many projects. Furthermore, you would need a lot of images to train the network from scratch.

Instead, it’s easier to use the pre-trained network and then use it to construct 128-d embedding for each of the 218 faces in our dataset.

Then, during classification, we can use a simple k-NN model + votes to make the final face classification. Other traditional machine learning models can be used here as well.

There are two main algorithms used in this project i.e. For face detection, and face recognition. Both algorithms are included in the OpenCv Python framework. The algorithm used for face detection is Hog Detector (Histogram of Oriented Gradients) which does two things:

Step 1: Converts the input image to black and white.

HOG only considers the changes between the light and dark areas in the image. It ignores the color information. That’s why it converts colored image into the black and white image.

Step 2: Looks for the gradient

Now after the step 1, it looks for the gradient in each pixel. A gradient is a direction from the lighter area to the darker area. It repeats the process for the entire pixels of a black and white image and draws the gradient image of it.

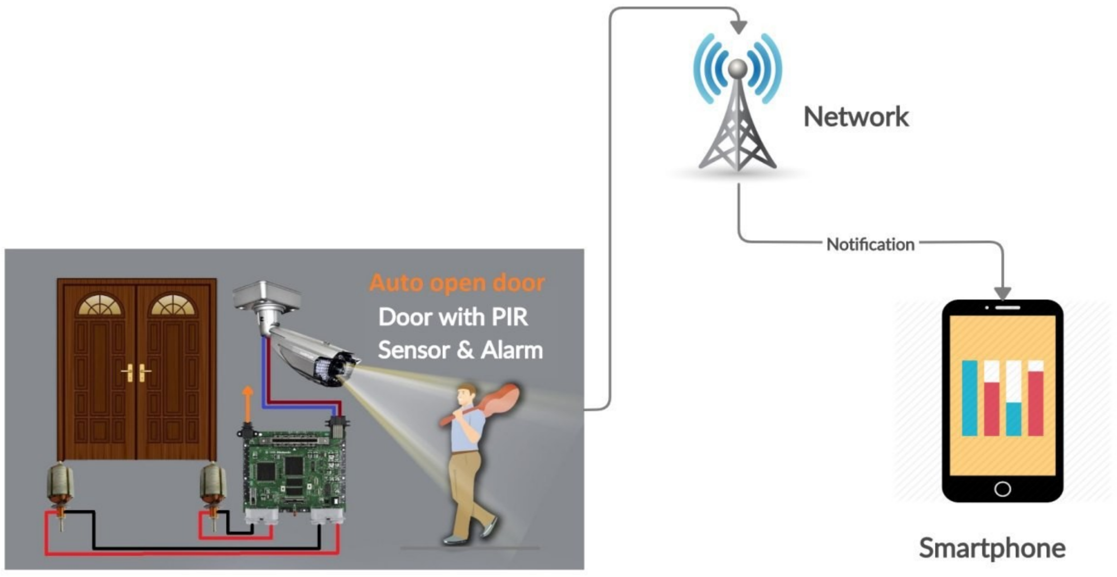
For face recognition, the face_recognition&dlib python libraries are used in the project:

The dlib library, maintained by **Davis King**, contains our implementation of “deep metric learning” which is used to construct our face embedding used for the actual recognition process.

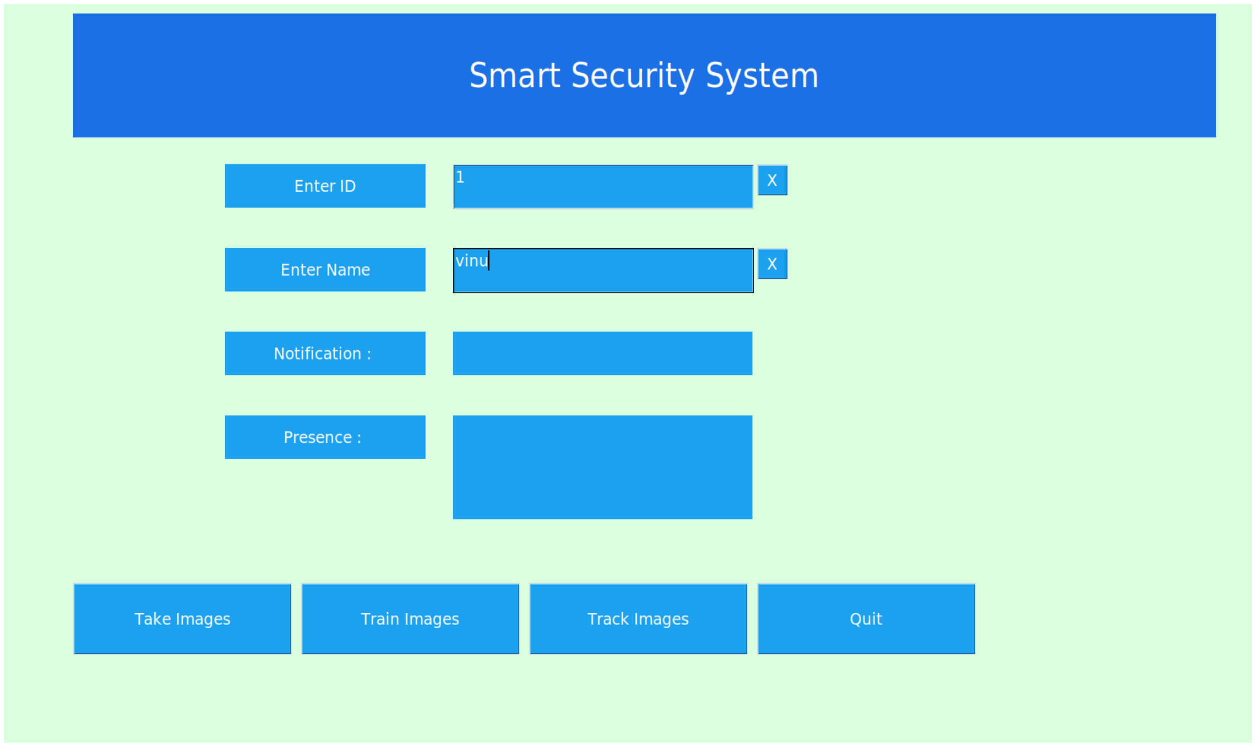
The face_recognition library, created by **Adam Geitgey**, wraps *around* dlib’s facial recognition functionality, making it easier to work with.

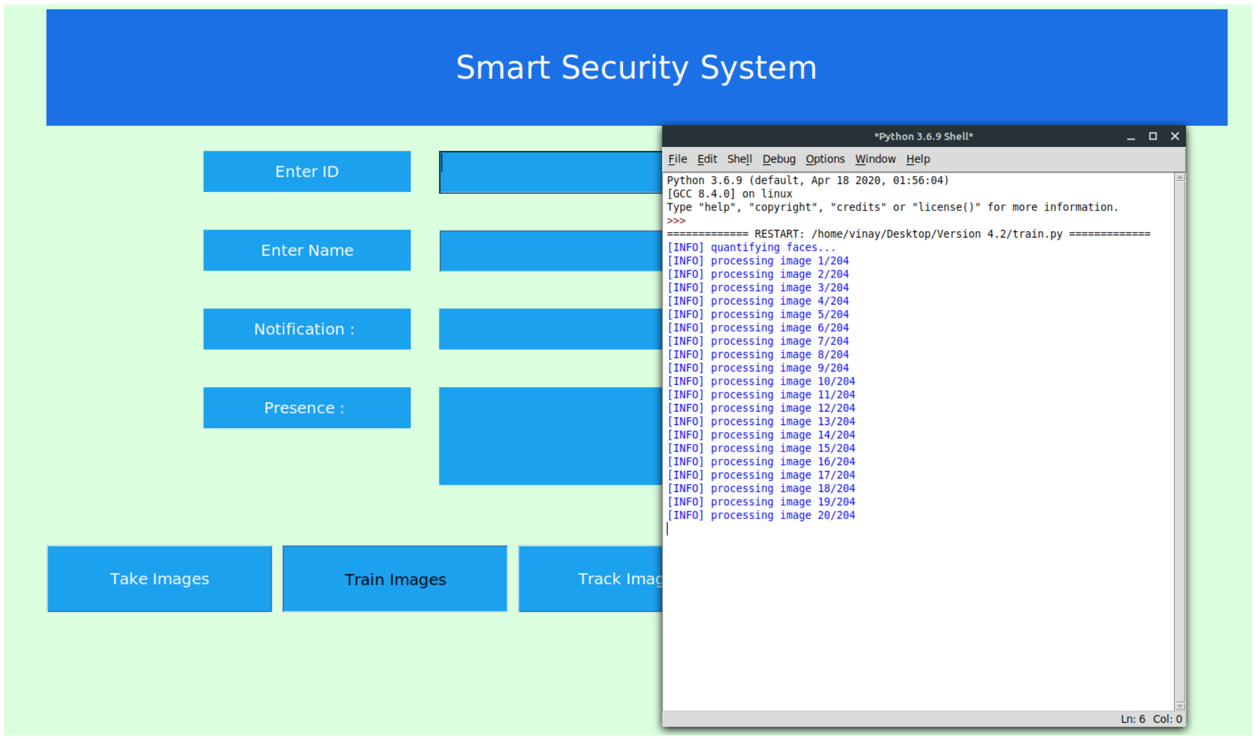
EXPERIMENTAL SETUP

The setup includes a security camera, PIR sensor, an alarm generator , and a door latch, connected to a Raspberry Pi board, owner's mobile phone. Once the PIR sensor detects motion, the camera activates, and starts detecting people and recognizing them, if the person is not recognized, the alarm rings, and a snapshot of the person is sent to the owner's mobile phone via internet. If the person is recognized, the latch gets enabled and they are allowed to enter through the door.

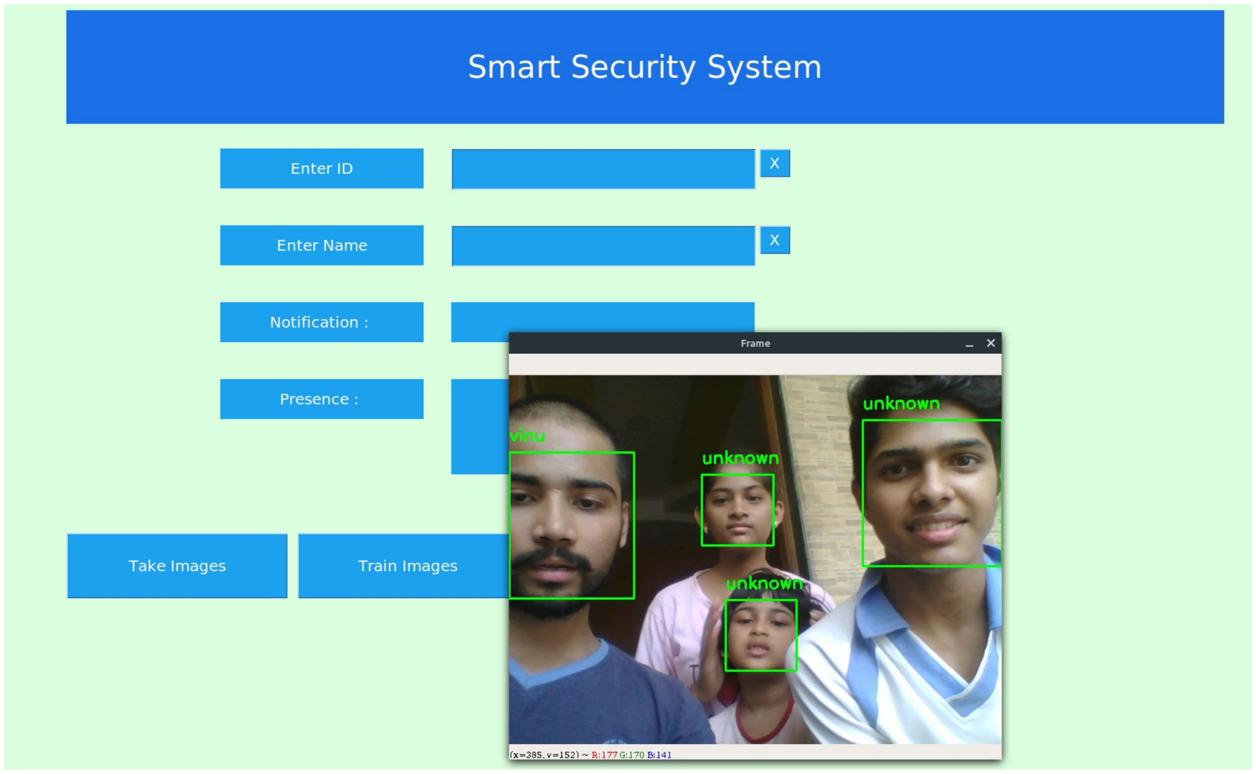


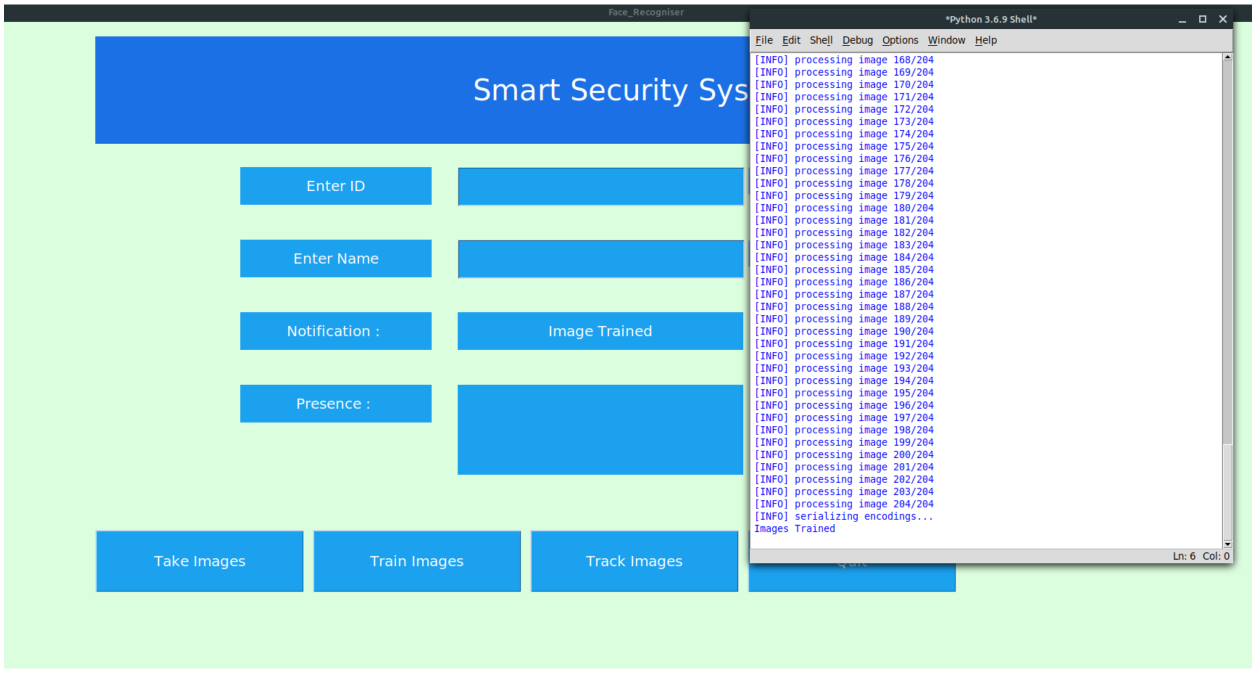
SNAPSHOTS



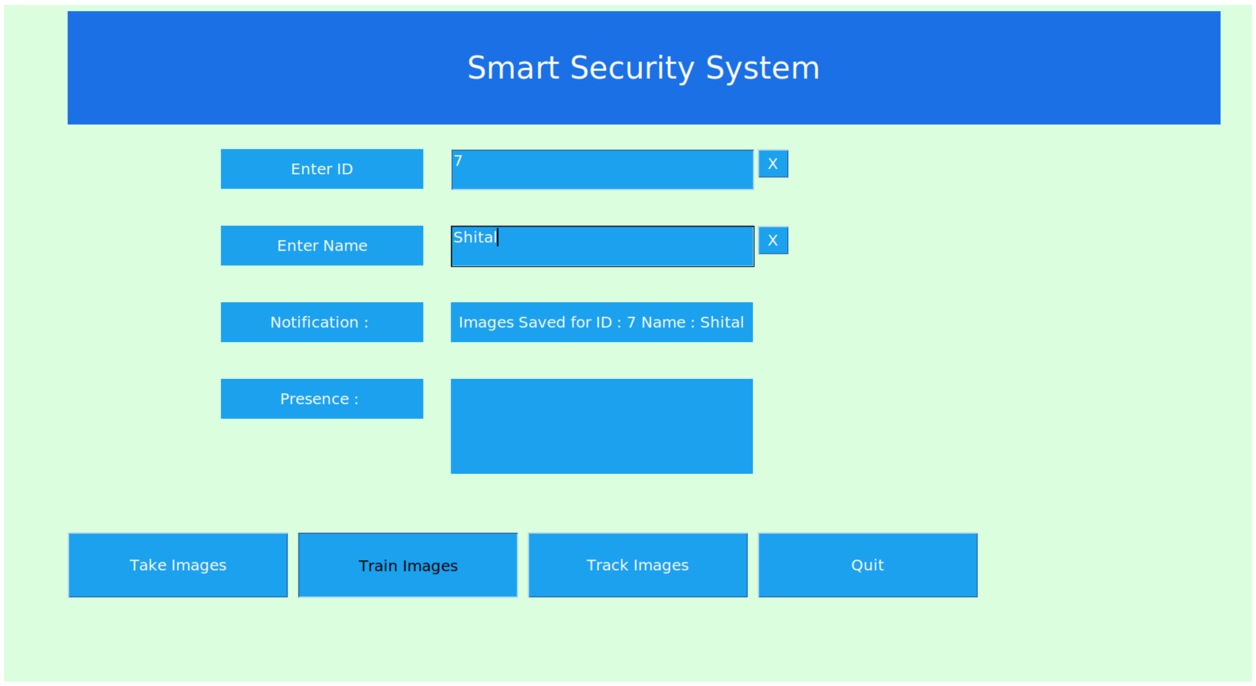


```
Python 3.6.9 (default, Apr 18 2020, 01:56:04)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============= RESTART: /home/vinay/Desktop/Version 4.2/train.py =============
[INFO] quantifying faces...
[INFO] processing image 1/204
[INFO] processing image 2/204
[INFO] processing image 3/204
[INFO] processing image 4/204
[INFO] processing image 5/204
[INFO] processing image 6/204
[INFO] processing image 7/204
[INFO] processing image 8/204
[INFO] processing image 9/204
[INFO] processing image 10/204
[INFO] processing image 11/204
[INFO] processing image 12/204
[INFO] processing image 13/204
[INFO] processing image 14/204
[INFO] processing image 15/204
[INFO] processing image 16/204
[INFO] processing image 17/204
[INFO] processing image 18/204
[INFO] processing image 19/204
[INFO] processing image 20/204
```





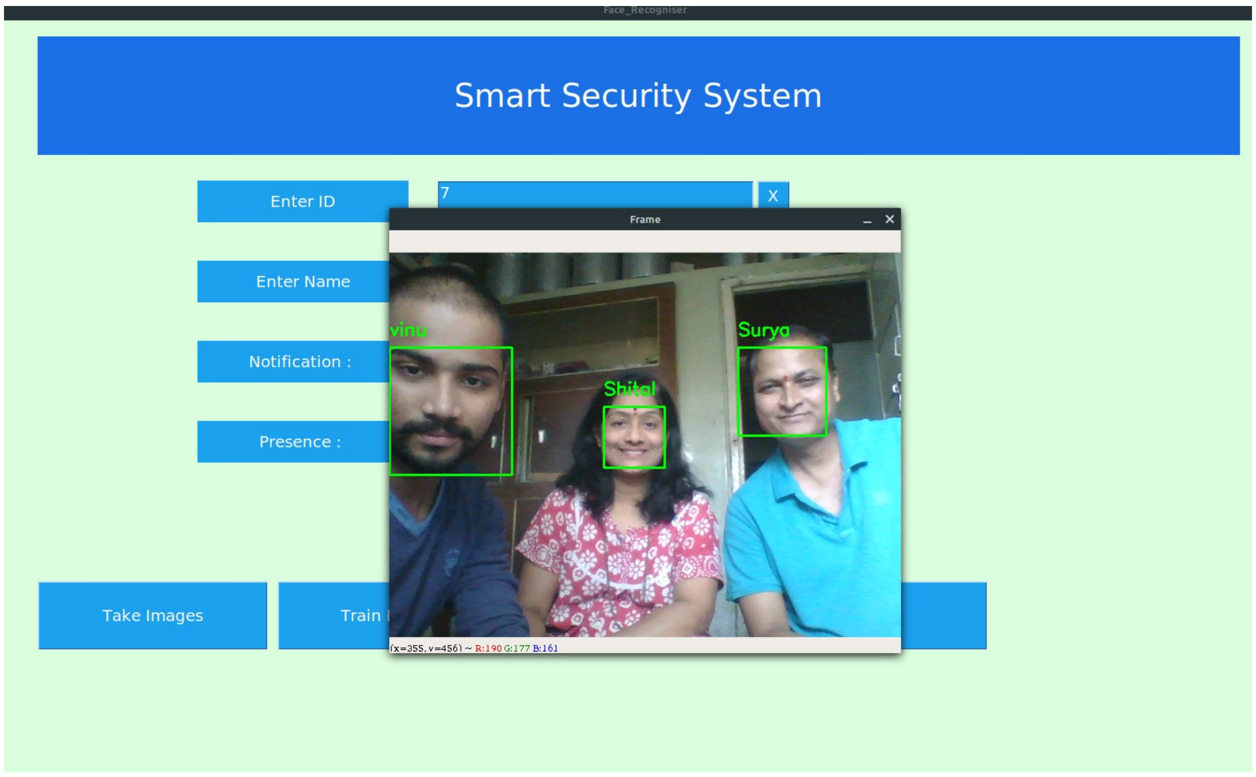
```
[INFO] processing image 168/204
[INFO] processing image 169/204
[INFO] processing image 170/204
[INFO] processing image 171/204
[INFO] processing image 172/204
[INFO] processing image 173/204
[INFO] processing image 174/204
[INFO] processing image 175/204
[INFO] processing image 176/204
[INFO] processing image 177/204
[INFO] processing image 178/204
[INFO] processing image 179/204
[INFO] processing image 180/204
[INFO] processing image 181/204
[INFO] processing image 182/204
[INFO] processing image 183/204
[INFO] processing image 184/204
[INFO] processing image 185/204
[INFO] processing image 186/204
[INFO] processing image 187/204
[INFO] processing image 188/204
[INFO] processing image 189/204
[INFO] processing image 190/204
[INFO] processing image 191/204
[INFO] processing image 192/204
[INFO] processing image 193/204
[INFO] processing image 194/204
[INFO] processing image 195/204
[INFO] processing image 196/204
[INFO] processing image 197/204
[INFO] processing image 198/204
[INFO] processing image 199/204
[INFO] processing image 200/204
[INFO] processing image 201/204
[INFO] processing image 202/204
[INFO] processing image 203/204
[INFO] processing image 204/204
[INFO] serializing encodings...
Images Trained
```



7

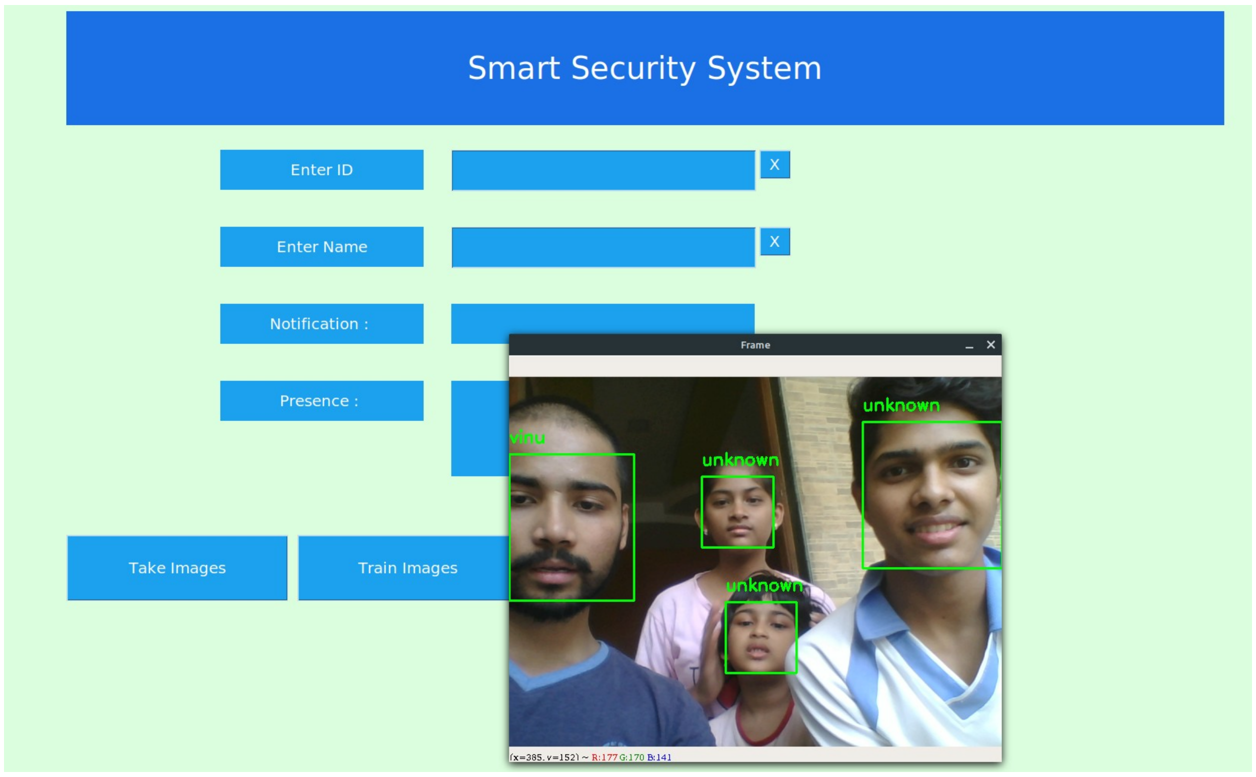
Shital

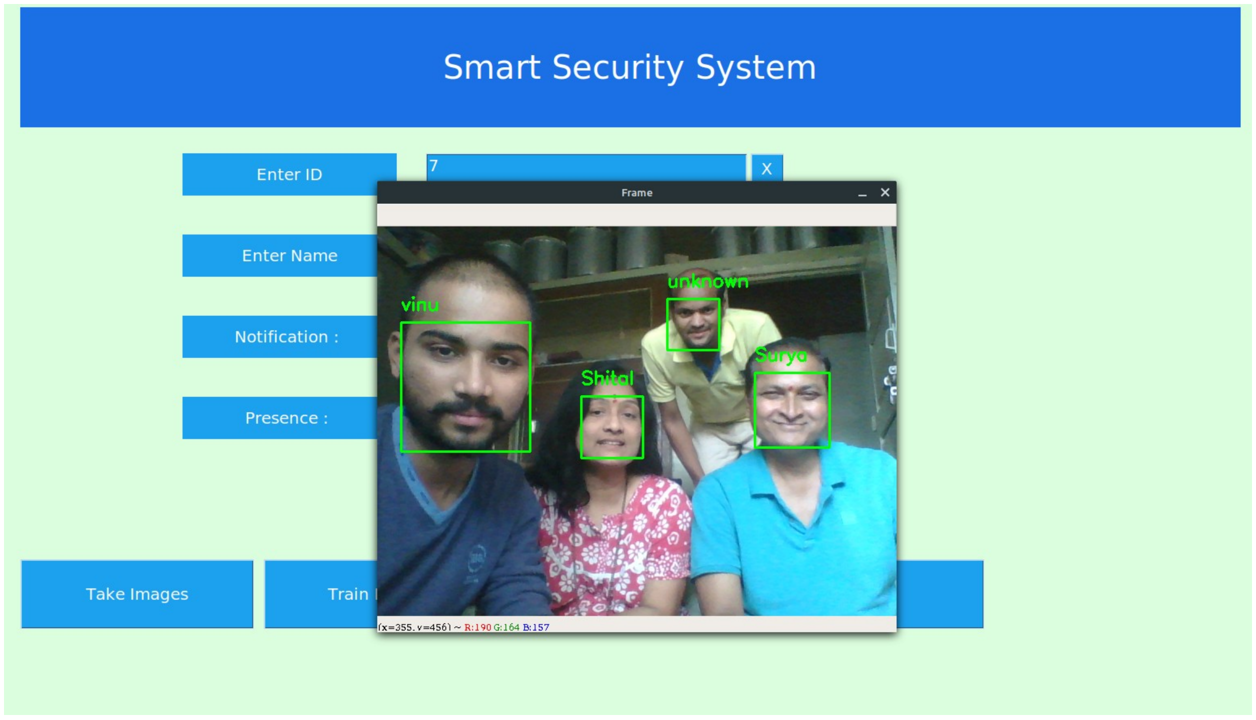
Images Saved for ID : 7 Name : Shital



7

(x=355, y=456) ~ R:190 G:177 B:161


Smart Security System
Enter ID
X
Enter Name
X
Notification :
Presence :
Take Images
Train Images
Frame
vinu
unknown
unknown
unknown
(x=385, y=152) ~ R:177 G:170 B:141


Smart Security System
Enter ID
7
X
Enter Name
Notification :
Presence :
Take Images
Train
Frame
vinu
unknown
Shital
Surya
(x=355, y=456) ~ R:190 G:164 B:157

TESTING

Test case No	Test case	Input	Expected Output	Actual Output	Status
01	Human Face	Image	Detected Face	Detected Face	Pass
02	Multiple Human Faces	Image	Detected face with name of that person	Detected face with name of that person	Pass
03	Multiple Human Faces	Image	All detected face with their names	Only one person identified	Fail
04	Multiple Human Faces	Image	All detected face with their names	All detected face with their names	Passs
05	Multiple Human Faces	Image	Show unknown tag for unknown person	Show unknown tag for unknown person	Pass

PERFORMANCE ANALYSIS

In the previous security camera system (CCTV), people could simply view and replay the videos captured in the CCTV cameras at an early point in time. There was nothing users could do if there was any uncertain situation that has happened earlier at the workplace. The action was taken after the incident has occurred.

In this project (Smart Security System), we have overcome this drawback and enhanced the security of the system. The user (owner) gets notified whenever anything uncertain happens or is detected in the security camera through multimedia message recorded, and an alarm is triggered and generated at the time of incident.

This gives user an idea of what is happening or going on while he/she is away from their place when they are not around. And also allows them to take action at that moment.

The people around also get alert by hearing the alarm and can know that something suspicious is happening nearby.

APPLICATIONS

1.Criminal identification

If FaceTech can be used to keep unauthorised people out of facilities, surely it can be used to help put them firmly inside them.

2. Security companies are using facial recognition to secure their premises.

3. IoT benefits from facial recognition by allowing enhanced security measures and automatic access control at home.

4. Bank Services

Banks use this product of Artificial intelligence as a security measure to detect any suspects entering without being identified.

ETHICS

Declaration of Ethics

As A Computer Science & Engineering Student, I believe it is Unethical To,

1. Surf the internet for personal interest and non-class related purposes during classes
2. Make a copy of software for personal or commercial use
3. Make a copy of software for a friend
4. Loan CDs of software to friends
5. Download pirated software from the internet
6. Distribute pirated software from the internet
7. Buy software with a single user license and then install it on multiple Computers
8. Share a pirated copy of software
9. Install a pirated copy of software

REFERENCES

[1]Sandesh Kulkarni, MinaksheeBagul, AkanshaDukare, Prof. ArchanaGaikwad “Face Recognition System Using IoT” (November 2017).

[2]Dr. PriyaGuptaa, NidhiSaxenaa, MeetikaSharmaa, JagritiTripathia” Deep Neural Network for Human Face Recognition” (January 2018).

[3]Numitha , Taha Noorain1, Amulya S Patil, Navyashree H, Ms. Nagalakshmi T S2,” Face Recognition and IoT Based Smart Lock Access System” (2018).

[4]Trung Nguyen, Barth Lakshmanan and Weihua Sheng “A Smart Security System with Recognition “(December 2018)