AI powered multi-level Arduino security system using facial and voice recognition

Makesh Srinivasan
19BCE1717

Naga Harshit
19BCE1547

Sandhya S
19BCE1614

May 2021

School of Computer Science and Engineering

Vellore Institute of Technology, Chennai

Keywords: Artificial intelligence, facial recognition, voice detection, gesture movement, machine learning, Arduino micro-controller and security systems

Abstract

Security has always been a domain of great importance. Throughout the existence of human beings, safety and security have accompanied growth and advancement. With advancement in technology and civilization, the need for security is only more essential, especially with the growth in artificial intelligence. Hence, we decided to combine both the domains and develop a system that can not only provide real-life use, but also help us learn various AI and ML algorithms, as we develop the product. We were able to produce a device that could provide three level security with facial recognition, voice detection and gesture movements.

Contents

1 Introduction 3

2 Literature survey 5

3 Methodology and Proposed work 7

3.1 Apparatus . . . 7

3.2 Data source . . . 8

3.2.1 Face recognition . . . 9

3.2.2 Gesture activation . . . 9

3.2.3 Voice detection/recognition source . . . 10

3.3 Data Pre-Processing . . . 10

3.3.1 Face recognition in an image . . . 10

3.3.2 Gesture detection . . . 11

3.3.3 Voice detection . . . 11

3.4 Machine Learning Model . . . 13

3.4.1 Encoding the face in the image . . . 13

3.4.2 Gesture detection . . . 13

3.4.3 Voice detection . . . 14

4 Results and observation 14

4.1 Gesture detection . . . 14

4.2 Speech recognition . . . 16

4.3 Face Detection . . . 16

5 Merits and Demerits 16

5.1 Merits . . . 16

5.2 Demerits . . . 18

6 Conclusion 18

7 Bibliography 18

1 Introduction

Security has always been an issue of interest; whether it is security of oneself and loved ones or security of information and data. Throughout the course of history, the life as we know it today exists only because of continuous persistence to survive, to adapt and to not be forgotten as a result of natural selection. Early humans devised clever ways to survive in the wild, they discovered fire and invented weapons while the rest of the animal kingdom continued to rely on sheer muscles, venom and other combat abilities. Homo sapiens dominate the planet today because they thought about protecting and securing themselves to ensure continued survival. However, life has always been dynamic - always in pursuit of change and evolution. In a world that is ever changing with the growth of digital and intelligence systems, it is difficult to refute that homo sapiens as we know it today will not exist in a few centuries. Contrary to what most popular movies depict, it is very unlikely for human beings to be invaded by aliens or be thrown over by super intelligent robots. A rational prediction would be that in future, human beings will merge with machines.

This statement may be perplexing or outright comical, but when given considerable thought about this, we realise that we have always been one with machines in the last few centuries, or at the very least, decades. Although machines as we know them exist only in the recent past, for the purpose of this claim, let us consider anything that is in-organic or an artificial extension of human abilities as "machines" (for the lack of better term, unnatural part of a human being). We rely on watches to keep us organised, to help track time. We use glasses or spectacles to enhance deteriorated vision. We use a faster and more effective medium of transport to travel. We use efficient weapons to control livestock or in extension, wildlife. We use computers to enhance our computational abilities. We are developing intelligent systems that are competing with humans, and sometimes even triumphing over us! History suggests that we are becoming more and more dependent on computers and machines; and this relationship will only continue to grow stronger and transcend from dependency to a symbiotic relationship - we will merge with intelligent systems to a point where it will be difficult to distinguish between human beings and intelligent machines. Our abilities will be enhanced and extended, for instance via exo-skeleton systems. Our brains would be connected and digitised, information can be uploaded and downloaded from the mesh network. In pursuit of increasing inter-connectivity and computational abilities, there is one domain that will always remain intact and increasingly significant and pertinent - security.

Regardless of the time period, security and safety will always be a crucial domain of concern. It is after all the basic driving force behind evolution - survival of the fittest. As aspiring computer science students, my team and I wanted to explore the domain of security from the perspective of computer systems. With keen interest in artificial intelligence (also the course of study), we decided to implement a smart security system for households.

Our aim is to ensure that our project can be used by most of the population and is implementable in the real-world. The simplicity of our project is very important because it allows us to learn about different algorithms and provides the freedom to try various approaches and learn more as we develop the project, something that a large scale project will not provide. Security in combination with artificial intelligence is a step towards mimicking human intelligence and in extension, an evolutionary step towards upgrading human potential.

In this J component, our aim is to perform facial recognition scan and if the person is authorised (present in the "Allow" database) then the access to the safe/room is granted, using an Arduino processor. Moreover, if the person is not authorised, the system will send a telegram message to the owner or the administrator via a Telegram-bot, and the owner can reply to "open" the door or ignore to keep the door closed. The administrator can add multiple-persons to the database and choose to allow access automatically, or even deny access to specific persons. The system will scan the

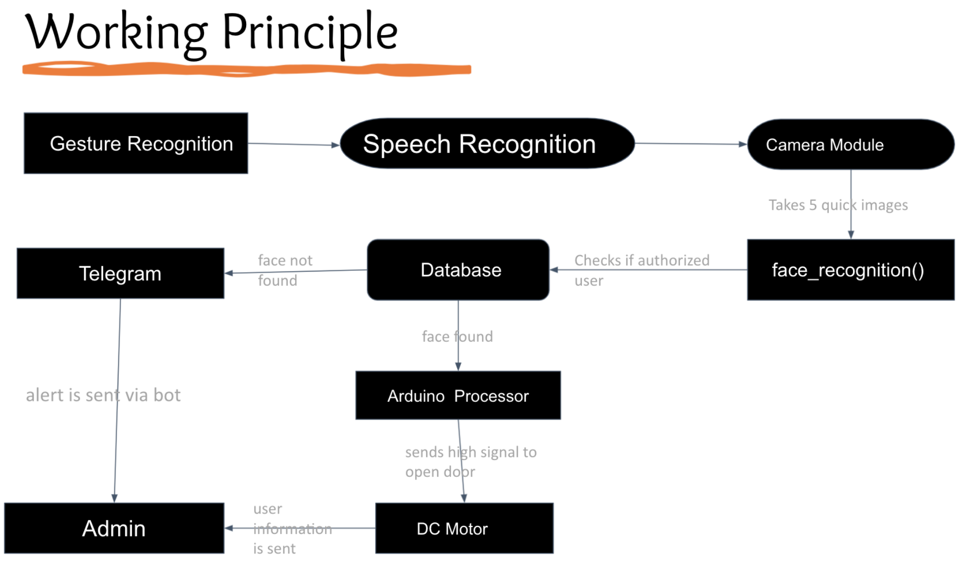


Figure 1: Workflow adopted in our research

face of the person as they walk to the room, and it checks the database if the person is authorised to enter the room. If the person is authorised to enter the room, the system sends a signal to the Arduino processor to open the door using a servo motor. Then a message is sent via Telegram to the owner that contains the name of the person. If the person is not authorised, a message is sent to the owner via Telegram to the owner, and the owner can reply to open the door or ignore it to keep it shut. This system works seamlessly with the Telegram application to enable the user to control the system easily and this is one of the reasons why we decided to keep Telegram as the interface instead of building an application for this system exclusively. This system needs to be able to detect accidental sensing, for instance, if a person walks too closely to the door and does not even intend to access the door, the system will think otherwise and scan the person's face and send a message to the owner. This can become annoying for the owner especially if the room is placed at a place where many people walk by it. To prevent this, we are incorporating a voice or gesture activated system; the person needs to ask the system to open the door by saying the command "Open door" (customisable by the owner). In addition to this, even a gesture activated system such as motion of the hand or waving at the camera can activate the machine to perform facial recognition. Both of these will reduce accidental activation.

Additional features such as setting a timer to always allow access and always deny access can be incorporated. For the set time period, the system will allow or deny access to everyone based on the conditions set by the owner. Provisions to change a user to administrator would also be included in the later stages, along with security alerts and warning messages for when camera lenses are blocked or power disconnected.

2 Literature survey

Automated AI based agents have made surveillance and private security to be very effective, cheap and efficient. This research paper [1] discusses the implementation of facial recognition security for doors locks. The system is implemented using the following open source technology :

1. OpenCV2 : It is an open source library of programming functions mainly aimed at real-time computer vision.
2. LBPH algorithm : Local Binary Pattern Histogram is a facial recognition algorithm which converts the features / landmarks of a face into a histogram, which is later used for classification.
3. SMTP : Simple Mail Transfer Protocol is a standard communication protocol to send Emails over the internet.
4. Raspberry pi3 : It is a microcontroller, which runs the script on its CPU and controls the hardware components.
5. pi camera : It is a hardware component connected to Raspberry pi which is used to take images of the user's face.

This system is designed to be used for basic security like home offices and campus.When the bell is triggered the image of the person is captured and this captured image is analysed with the libraries and algorithms mentioned above. A captured face is compared with the data set and if a matching face is found the access is granted and the door opens. Else the access gets denied and the captured image is sent to the owner using the SMTP protocol to their email. A preset time is fixed in the system. The system waits for this amount of time for the owner to respond. The response is retrieved on raspberry pi using IMAP protocol. The retrieved message is analysed and the user is either denied or granted access. Wireless communication is achieved using SMTP and IMAP.
The motivation behind this research paper [2], is to implement a real time solution for home security problems. The aim of this paper is to provide an efficient, cost-effective system. It also provides real time solutions for Home Security with Raspberry Pi based face recognition system. Developing a computational model for face recognition is a difficult process because a person's face is a complex multidimensional visual model. Here they have presented a methodology for face recognition based on the information theory approach of coding and decoding the face image. The Eigenface approach using the Principal Component Analysis(PCA) algorithm is used for facial recognition. Conventional face detection techniques such as Haar detection and PCA are implemented in this face recognition system. The aim of this system is to replace the use of ID cards, smart cards or Biometrics with face recognition to enhance the security. The uniqueness of each face is detected using the vector numbers in the Eigenface approach. In this approach each face image is approximated using a subset of the eigenfaces, those associated with the largest eigenvalues. The neural network for facial classification and recognition is also represented in this paper. LCD, Camera, Motor and Raspberry Pi board are the components used to build the hardware of this face recognition system.
Home security has become a crucial part of society [3] this research paper has explored the use of face recognition as bio-metric security. Face recognition is more stable among other bio-metric identification methods as it is using the human face that results in high accuracy, lowest false recognition rate. Thus, this method is a much more practical security system compared to other security systems. Face recognition is likely the most natural approach to perform bio-metric verification between individuals. Face detection is the first step in the face recognition process. This feature

enables recognition of an individual without a physical contact with any hardware device(much needed feature in this COVID situation). Many techniques such as Principal Component Analysis(PCA) can be used for face recognition. PCA is one of the most popular techniques used for face recognition. This technique transforms a number of correlated variables into a number of uncorrelated variables mathematically. PCA technique utilizes the use of Eigenfaces for face recognition as it is effective and an efficient way to represent pictures into Eigenfaces components. This is so because it can reduce the size of the database of the test image. In order to improve the performance of face recognition technology many new features and methods are being developed and deployed. Deep learning techniques are used in face recognition technology. It is an extensive group of machine learning algorithms which is based on learning data representations, as opposed to task-specific algorithms. Deep learning can either be managed or semi-directed or unsupervised. With deep learning, the system is improved from time to time. Some images of authorised users are stored in the system's database. The system uses these images for training newly captured faces during the face recognition process. Basically a large scale comparison takes place between the authorized image stored in the database and the image captured through the web camera. This way the accuracy of the system is increased on a large scale. Home security is an example of an Internet of Things (IoT) application. IoT is a technology where devices and the internet are interconnected. IoT refers to the network of associated physical objects that can interact and trade information among themselves without the need of any human intervening. By using IoT, it can help in controlling door access and also send notifications throughout the internet. Many researchers choose to use an embedded device called a Raspberry Pi for training and identification purposes. The fundamental reasons behind picking this component is its high handling limit, low cost and capacity adjustments in various programming modes.

This paper [4] is more focused towards the purpose of Smart Door Unlock System based on Face Detection which enriches the security system on a larger scale. With this system one of the major limitations of dealing with keys is resolved. Though this system does not ensure 100% safety, it makes the system key-less which is one of its biggest advantages. This paper proposes a machine learning approach with Haar Cascade method. Here a camera sensor is used to capture the face and an image matching algorithm is used to detect the authenticated faces. Only the person whose face is matched can be able to unlock the door. Many types of door automation were developed such as smart cards fingerprint access but they weren't prominent enough. Hence face recognition based automation became the best solution.Though this system turns out to be the best system it has its own pros and cons. Some of the cons in this system are the lighting issues and background environment issues. One of the main pros of this system is acquiring the door using a face detection approach and the entire face is recognized. An intense innocence is expected in the security industry which is achieved by attribute extraction from facial image with the help of smart door models in the face recognition technique. This system will be robust from hacking attacks as the proposal of this system is based on a machine learning approach.

This paper [5] concentrates on one of the quickest human computer interaction methods which led to the trend "robotics", which also includes interaction with robots with human expressions. Face detection, skin color detection, and image processing are some parts of the face recognition system. Bi-linear interpolation and improved linear discriminant analysis are the two main methods of face recognition. Interaction with robots through facial expressions is the most direct and rapid method of human-machine interaction. Pre-establishing facial expressions, reducing noise data through cross validation, using a vector pursuit to increase the speed of facial expression classification machine training are the prominent ways through which the system recognizes the facial expressions that couldn't be recognised earlier. One of the face recognition features that this paper mainly focuses on is Bilinear interpolation. Bilinear interpolation reduces the face image to a narrow-diagram which facilitates discriminant analysis computing. Another feature that this paper focuses on is skin color detection. The skin color blocks are picked and it gets inputted as an image which gets converted into binary format. Drawback hits when the background of the captured image is complicated. So morphologic processing is needed to make the binary image pixel erosion and dilation, this way, the mottle in skin color blocks would be reduced and internal pixels would be dilated.

This paper [6], focussed on improving the security by using a smart locking mechanism that opens and closes based on who is trying to gain access. The paper is focused towards aligning objectives and creating an intelligent system that recognizes people and also increases the ease of interaction and communication via telegram application. The telegram application comes into use when a person is not authenticated automatically by the system. Histogram equalisation and fisherface method is used in the final sections of this paper. The face is one of the most frequently physiological parameters used to determine the identity of each individual. An image is acquired via an input module (security camera/ web camera) then it gets processed through a PC. When the person is authenticated/recognised by the system, the microcontroller (Arduino) sends a signal to the servo motor to open and close the doors. This way the security is improved and the automation of some trivial processes are adapted. An accuracy of 83% was produced and this could be improved further by extending the research done by the Hangzhou Normal University, China, as they explored more parameters such as gender, skin tone, etc.

3 Methodology and Proposed work

Python cv2 library has been used to get images directly from the camera or webcam module. When a face is detected the camera module takes 5 images, this is done so to avoid any blur or unclear images. The locations of the nose, eyes, mouth and chin are detected using the face_recognition library along with openCV.

Once a face is recognised in an image, it is highlighted using a green square, the image is then converted from the BGR color format that openCV uses into the RGB color format that is required for face recognition. The images are converted to face_encodings using the RGB color image and locations of nose, eyes, mouth and chin.

The images present in the database (allowed users) are also converted to face_encoding using the same above steps, the images from the database each have a label, i.e, the name of the person in the image.

The face_encoding from the camera module and from the database are compared to check if any matches are found, if a match is found, the user's name is displayed below the green box used to highlight the face.

Once a user has been recognised, the python script sends and sms to the owner that a particular person has been allowed inside, this is done using the Way2sms API, it also sends a command to the Arduino which in turn sends commands to the servo motor that opens the door lock for the user.

If no matches are found from the face_encoding, the owner is alerted with an SMS message using the Way2SMS, an image of the unknown user is also sent as a telegram message to the owner. The owner upon inspection can choose to open or keep a door closed using simple telegram commands.

3.1 Apparatus

1. Hardware

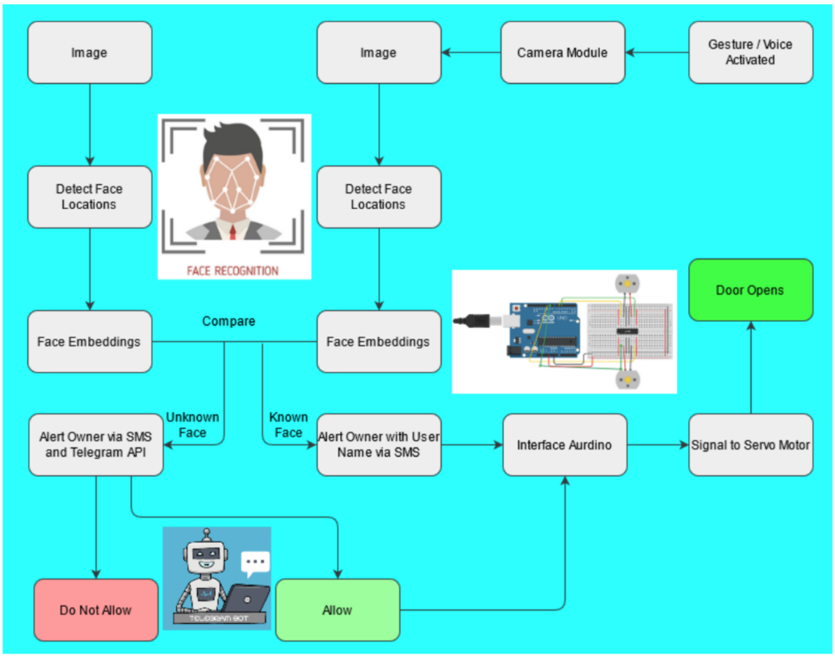


Figure 2: Workflow adopted in the process

(a) Arduino
(b) Servo motor
(c) L239D IC
(d) Web camera
(e) Microphone
(f) Laptop/personal computer
(g) Connecting wires
(h) Mechanical arm - to open door
(i) Media cable - to connect computer with Arduino processor

2. Software

(a) Python 3
(b) Telegram
(c) Some ML and facial, voice and gesture recognition libraries
(d) Arduino IDE
(e) Editor

3.2 Data source

This section is divided into three parts: face recognition, gesture detection and voice detection.

3.2.1 Face recognition

The images in our database are images of registered users, only the owner can add or remove from their database. All the users in this database will have an image with a label which is the user's name. Only these users will be allowed to access the door. The camera module is activated when the gesture or voice message is provided by the user, it takes 5 images of the person at the door and passes the data to the python script for facial recognition. Five images are taken so as to avoid blurry or unclear photos where the face cannot be detected clearly or can be mistaken for someone else's face. Facial recognition can be performed in many ways. In openCV, the images can be uploaded and the models can be trained from scratch, using Neural Network or Hidden Markov Models. In case of a NN, a multi layer perceptron is used to perform analysis and make decisions in each layer. The feature vectors are obtained using Gabor wavelength transforms. Hidden Markov Models are a statistical tool used in face recognition. They are used in conjunction with neural networks. It is generated in a neural network that trains pseudo 2D HMM. The input of this 2D HMM process is the output of the ANN, and It provides the algorithm with the proper dimensionality reduction[7]. Building models from scratch will be time consuming and we simply do not have the resources to perform high quality neural network modelling as our laptops and personal computers offer GPUs and CPUs that cannot handle such computations in a reasonable amount of time. Moreover, the major disadvantage of using such models is that the speed of detection will be very low. The slow detection and interaction between the system and the users will render the whole security system moot as the most important aspect of security is agility, speed and precision. The Haar-cascade classifiers is an effective object detection method that performs classification quickly and relatively precisely. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. The algorithm uses haar wavelets that consist of rescaled square shaped functions that represent eyes, nose and other parameters (line and edge features). The algorithm is trained using positive and negative images. Positive images are those images that contain the subject - in our case, it is the face. Negative images are those without the faces. Once trained, models can detect the faces with an accuracy of 95% with even 200 features. However, the model resulted in 16,000 features, even though the accuracy is much higher now, this makes the process very slow; however, the developers reduced it down to 6000 by using a clever technique called as Cascade of Classifiers, where the algorithm performs search on sub windows every instant instead of looking at the whole window at a time as there are higher chances of not being able to find a face. According to the developers, their detector had 6000+ features with 38 stages with 1, 10, 25, 25 and 50 features in the first five stages, and the use of windows allowed for much faster screening and removing the parts of the image that do not contain the subject (face). We decided to use this model (haarcascade_frontalface_default) for detecting faces in combination with voice and gesture recognition algorithms as this is faster and much more efficient. The accuracy of this model competes with that of the most complex algorithms yet provides a fairly quicker detection mechanism.

3.2.2 Gesture activation

The system is activated (the scanner starts scanning when actuators - voice or gesture - initiate the process). Gesture activation is performed using some hand symbols, and in theory, they can be anything as long there are enough images in the dataset to train. However, we as three students are not equipped with man-power to conduct primary data collection of hundreds or thousands of images of various hand gestures, we rely on what is already available. American sign language (ASL) is a language used by people who are unable to speak to communicate. They use special symbols to indicate different letters of the english alphabets and even other common terms such as left and

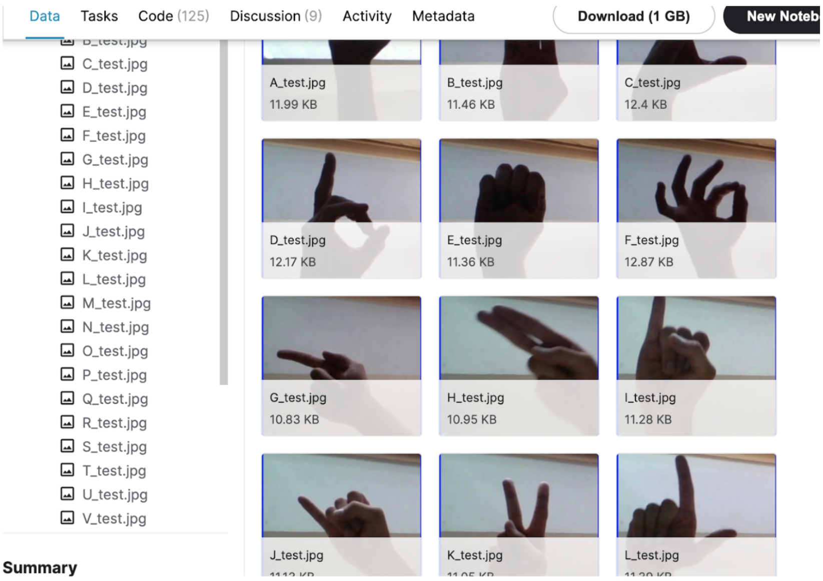


Figure 3: Dataset ASL hand gestures

right. The dataset that we found here [8] on Kaggle provides 1GB of data on different symbols shown on the hand under various environmental conditions such as dark room, light surroundings, and with high noise and distortion. The image shown below contains a few examples of the images in the dataset. It would be interesting to customise the hand gestures to something of personal interest such as the spider-man hand sign for web shooting. However, to customise the hand gesture we need enough data for training and we do not have a readily available dataset for the signs that the user/owner may wish to use. Hence, we are resorting to ASL.

3.2.3 Voice detection/recognition source

Speech recognition in general terms can be treated as a structured search problem. Correct / accurate recognition can be defined as outputting the most likely word sequence given the language model and the acoustic model. Speech recognition involves database creation, training a model using the database and using the trained model to predict/recognize speech. Database creation describes the collection of speaker's voice samples and extraction of features for selected words. One such Dataset in TensorFlow's Speech Command Dataset. It includes 65,000 one-second long utterances of 30 short words, by thousands of different people.

3.3 Data Pre-Processing

This section is divided into three parts: face recognition, gesture detection and voice detection.

3.3.1 Face recognition in an image

Face detection can be thought of as such a problem where we detect human faces in an image. There may be slight differences in the faces of different humans but on a whole, we can assume that there are certain features that are associated with all the human faces. These features include nose, eyes, mouth and chin. Here we use the face_locations function to find the area of the face in the images taken by the camera module. This function takes in 3 paraments [11],

1. Image - Location or directory of the image can be provided
2. Upsample - this parameter refers to the number of times the process must be repeated on the face, the more number of times it is repeated the more faces it can find in an image. It is usually set high for images where small faces in the background must be detected. By default it is set to 1, meaning it will detect the nearest face to the camera.
3. Model - This parameter refers to the face detection model that needs to be used. It can take two values
   (a) "hog" is less accurate but faster on CPUs.
   (b) "cnn" is a more accurate deep-learning model which is GPU/CUDA accelerated (if available). The default value for this parameter is "hog".

3.3.2 Gesture detection

As for detection of the hand gestures, there are 3 main extraction techniques

1. Histogram of Gradients (HOG)
2. Principal Component Analysis (PCA)
3. Local Binary Patterns (LBP)

Once the extraction is done using all three techniques, we will select the most efficient and fastest algorithm and implement various machine learning algorithms on the dataset (Random Forests, Support Vector Machines, Naïve Bayes, Logistic Regression, K-Nearest Neighbours, Multilayer Perceptron).

The [9] first step is to perform segmentation and convert the RGB image to grayscale image of a single channel. Once converted, we will use Canny edge detection to reduce the background noise by determining the edge or the border of the subject. It uses a multi-stage algorithm to distinguish sharp discontinuities or edges. This concludes the first stage of gesture detection.

In the next stage, we perform feature extraction to determine the 32 dimensional vector for the test sample. Followed by this, Generation of Histogram of Visual Vocabulary is generated. Finally, we perform classification, and this is discussed more in detail in the Machine learning model section.

3.3.3 Voice detection

Feature extraction for the speech samples in the database is done using Mel Frequency Cepstral Coefficients algorithm (MFCC). This algorithm uses linear spaced filters and logarithmically spaced filters to capture the important features/characteristics of speech. The result of applying the MFCC algorithm is a set of coefficients called acoustic vectors , i.e, each input utterance is converted into an acoustic vector. Speech recognition requires a model for prediction of the words uttered in the speech, one such model can be built using the Hidden Markov Modelling Approach (HMM). [10]
HMM creates stochastic models from known utterances and compares the probability that an unknown utterance was generated by each model. The acoustic/feature vectors are arranged into a Markov matrix that stores the probabilities of state transitions. Meaning if each code word were to represent some state, HMM would follow the sequence of state changes and build a model that includes the probabilities of each state progressing to another state.

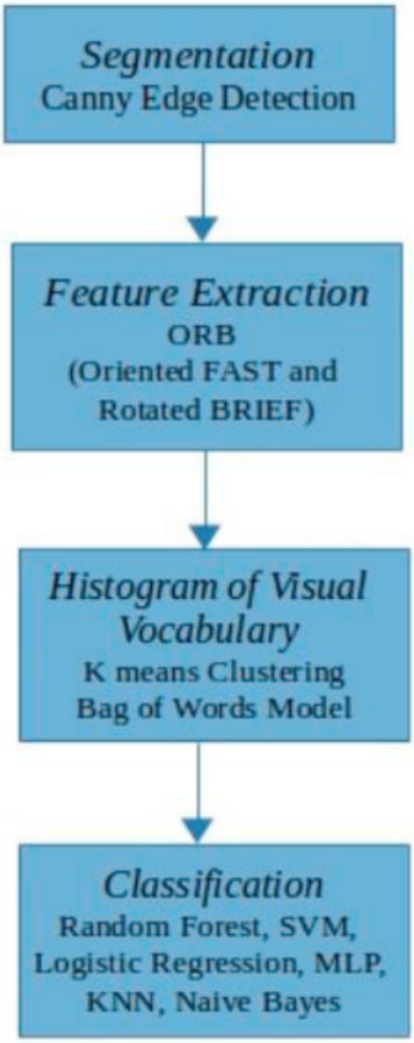


Figure 4: Workflow of gesture detection

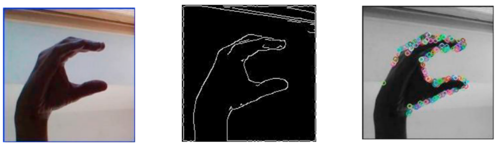


Figure 5: Feature extraction stages

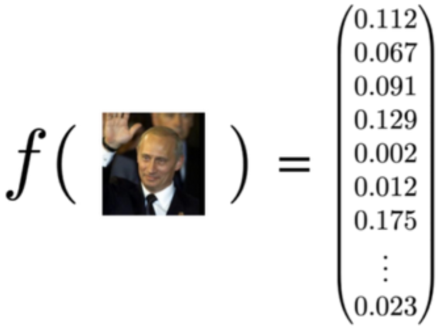


Figure 6: Encoding of a face

3.4 Machine Learning Model

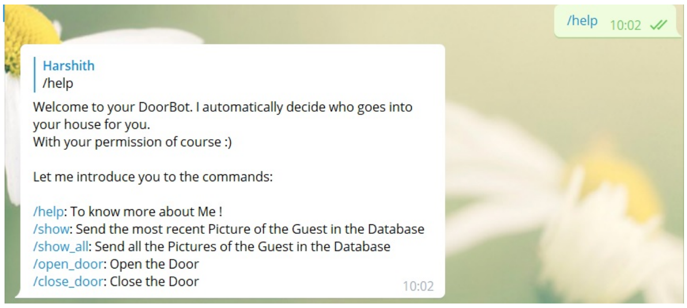
3.4.1 Encoding the face in the image

Encoding the face can be thought of as converting the face of a person (an image) into numbers. Since we have found the location of the face in the image in the previous step we can use this to extract features from it. Face embeddings allow us to extract the features out of the face.

A neural network takes an image of the person's face as input and outputs a vector which represents the most important features of a face. In machine learning, this vector is called embedding and thus we call this vector as face embedding. The neural network is trained such that the network learns to output similar vectors for faces that look similar. For example, if I have multiple images of faces within different timespan, of course, some of the features of my face might change but not up to much extent. So in this case the vectors associated with the faces are similar or in short, they are very close in the vector space. Training a neural network takes a significant amount of time and thus is not feasible in real time applications, thus for our project we will be using a pre trained neural network, which has been trained using over 3 million images. This helps us reduce the computation speed drastically and provides the user at the door with a quick response.

3.4.2 Gesture detection

For the gesture detection component of the project, we use Random Forests, Support Vector Machines, Naïve Bayes, Logistic Regression, K-Nearest Neighbours, and Multilayer Perceptron and determine the accuracy and error percentage of the models. As of this review, we have not created the models for each algorithm but the plan is to take the output of the Generation of Histogram of Visual Vocabulary stage in Data-preprocessing 4.2 and use it as input to classify the vector representing the hand gesture symbol into an english alphabet. For instance, if the character is "L", then the vector will be generated in the preprocessing stage and this will be used to train the model and teach what the letter "L" will look like in the 32 dimensional vector form. Then the test sample image is taken and preprocessed to get the 32 dimensional form again, and the classification begins. For each algorithm, the process is different, in general, the output will be the letter and the overall confidence in the prediction. Based on the accuracy rates and error rates from this stage, we will determine the best model and use that in the later stages of our project, in combination with face



/help 10:02

Harshith
/help
Welcome to your DoorBot. I automatically decide who goes into your house for you.
With your permission of course :)

Let me introduce you to the commands:

/help: To know more about Me !
/show: Send the most recent Picture of the Guest in the Database
/show_all: Send all the Pictures of the Guest in the Database
/open_door: Open the Door
/close_door: Close the Door 10:02

Figure 7: Telegram functions

recognition and voice recognition.

3.4.3 Voice detection

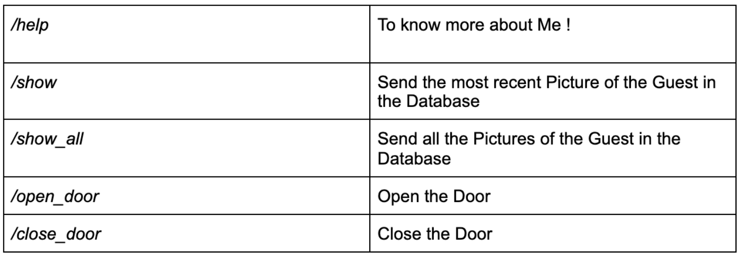
Feature extraction and model building are the basic components of speech recognition [10]. The search process is the most important part of speech recognition. The search algorithm computes the likelihood of the observed data by scoring it on the feature models. It then chooses the speech pattern with the highest likelihood. The number of possibilities is endless, thus it is crucial to use a good search algorithm that can increase efficiency and accuracy and reduce computational time to the maximum. One such algorithm is The Viterbi algorithm. It is a dynamic programming algorithm for obtaining the probability estimate of the most likely sequence called the viterbi path. It imposes the restriction that the cost, or probability of any path leading can be recursively computed as the cost in making a transition from the previous state to the current state. In short, the algorithm finds all possible end states with similar sequence to the input speech and finds the end state with the maximum score. It then backtracks through the graph to find the most probable sequence of words.

4 Results and observation

Telegram Bot allows the user to connect with the Arduino remotely. Users can execute various commands in the bot to perform necessary actions. The /help command provides a list of all commands that exist and gives a brief description about each command. A warning message containing the name of the user who has accessed the door is sent to the Owner via Telegram Bot every time the door is used. The Owner can use the command /show to view the most recent image taken by the camera module. The /show_all command sends all the images taken by the camera module. /open_door command sends a signal to the Arduino to open the door, similarly /close_door sends a signal to the Arduino to close the door.

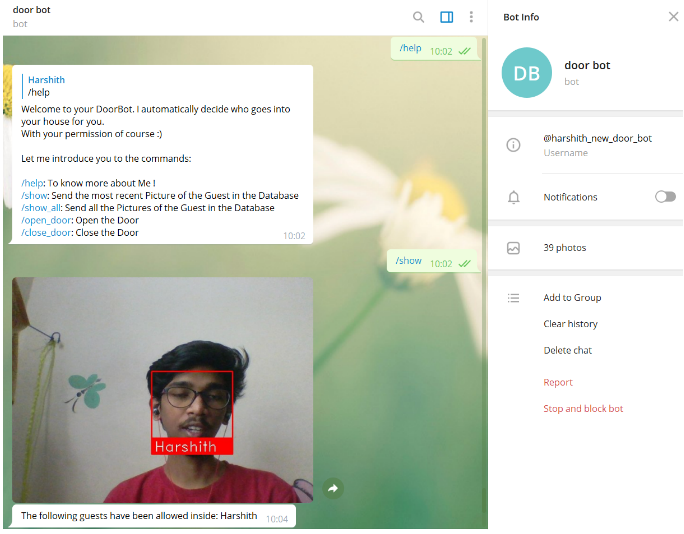
4.1 Gesture detection

Gesture Detection uses the camera module as a video source. It generates a black and white video using the source by converting every pixel that is of skin color to white and any pixel that is not skin color is converted to black. This generates a layout of the hand, which is used to identify how many fingers are raised or what shape is being shown. If the necessary sign or shape is shown for a set period of time, the gesture detection returns true and speech recognition is executed.



/help	To know more about Me !
/show	Send the most recent Picture of the Guest in the Database
/show_all	Send all the Pictures of the Guest in the Database
/open_door	Open the Door
/close_door	Close the Door

Figure 8: Telegram functions


door bot
bot

/help 10:02

Harshith
/help
Welcome to your DoorBot. I automatically decide who goes into your house for you.
With your permission of course :)

Let me introduce you to the commands:

/help: To know more about Me !
/show: Send the most recent Picture of the Guest in the Database
/show_all: Send all the Pictures of the Guest in the Database
/open_door: Open the Door
/close_door: Close the Door
10:02

/show 10:02

The following guests have been allowed inside: Harshith 10:04

Bot Info

door bot
bot

@harshith_new_door_bot
Username

Notifications

39 photos

Figure 9: Telegram UI

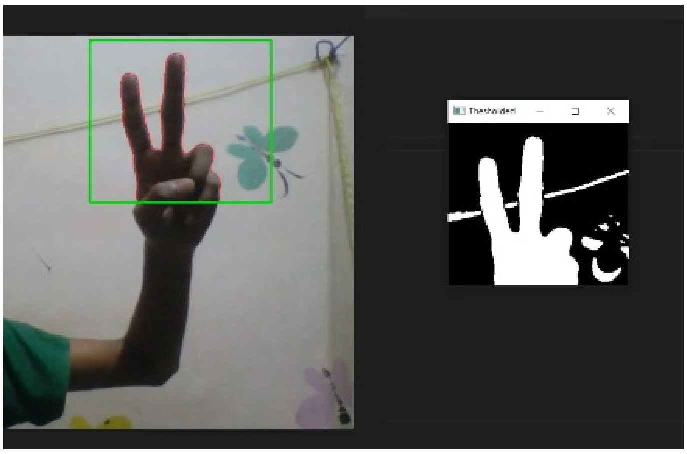
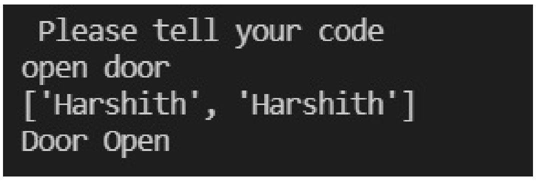
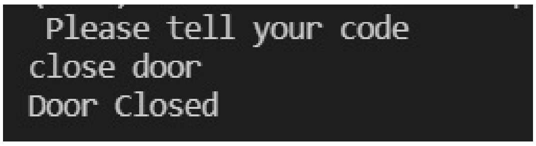


Figure 10: Gesture recognition - peace sign for unlock



```
 Please tell your code
open door
['Harshith', 'Harshith']
Door Open
```

Figure 11: Facial recognition tested



```
 Please tell your code
close door
Door Closed
```

Figure 12: Door opened when recognised closed when not recognised

4.2 Speech recognition

The speech recognition module is executed using the speech_recognition library in python. Initially it listens and calibrates the noise levels so it can clearly distinguish between the user's voice and background noise. Once an audio is heard, it uses the speech to text conversion provided by Google to convert the recorded audio into text. The returned text is compared with the password, if both match the door is opened and if they don't match the process restarts again from gesture detection. Different passwords can be set to achieve various activities.

4.3 Face Detection

The face detection module is executed if and only if speech recognition and gesture detection are completed successfully. The images of users who are allowed to access the door are taken from the database and converted into image encodings using the face_encodings( ) command. These image encodings are stored as a list with the name known_face_encodings. Five images are taken to avoid redundancy using the camera module. The images are stored in the database to be used for detection and to send images via telegram bot. OpenCV is used to resize the images and convert them into RGB frames since the face_recognition library uses this format. The RGB frames are passed to face_locations( ) to identify the locations of the face in these images, the image is then converted to encodings using the face_encodings( ) command. These encodings are compared with known_face_encodings to identify the person who is at the door. If only a known face is identified, the user is allowed to access the door and a notification is sent to Owner via the Telegram Bot that a person has used the door. If all the faces detected in the image are Unknown faces, a warning message is sent Owner via Telegram Bot and the user isn't allowed to access a door.

5 Merits and Demerits

5.1 Merits

1. The model used to detect and analyse faces is fairly accurate and thus can increase Security levels significantly.

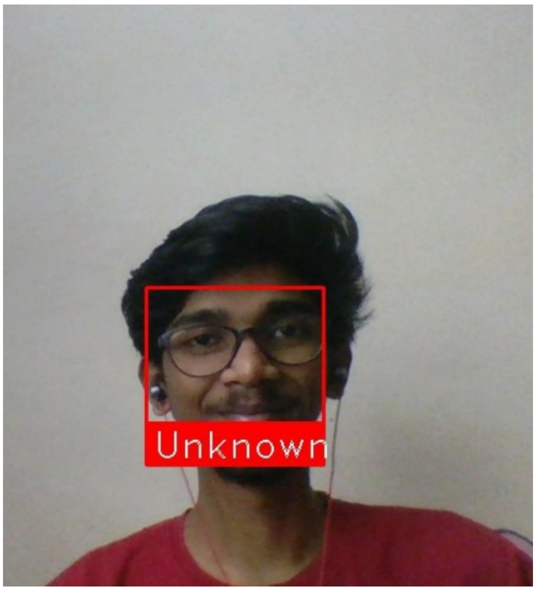


Figure 13: Unrecognised face

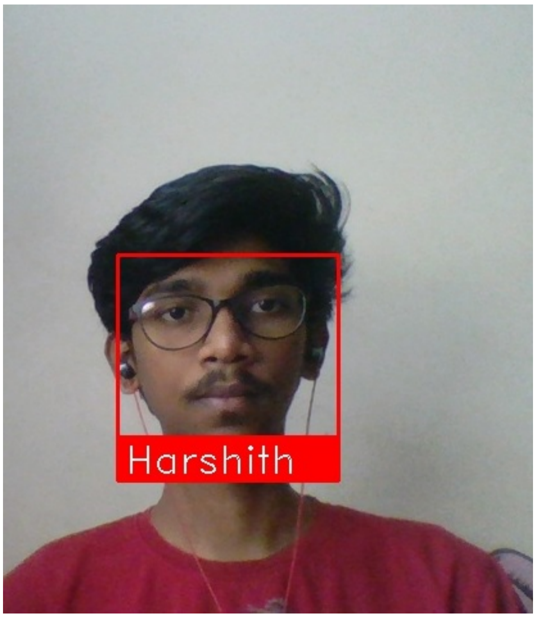


Figure 14: Recognised face

2. High accuracy allows avoiding false identification.
3. As the system runs using an Arduino, it is completely automated and requires fairly less power supply to run.
4. OpenCV libraries used are much faster and efficient when compared to other libraries and thus can provide fast and accurate results even on a low end system.
5. Using a pre-trained Neural Network model also reduces the time and space complexity drastically.

5.2 Demerits

1. When multiple faces are seen in the image and supposed only one or few of them are authorised and the rest are un authorised, the system does not know what to do.
2. If the person is standing too far away from the camera, the face recognition algorithm cannot clearly identify if the image is a match or not.
3. Gesture detection may not work if the user is too far away from the camera or too close to the camera. User needs to stand at an optimal distance.
4. Speech detection may not be able to interpret the speech if there are multiple voices or the user's voice is muffled due to noisy background.
5. In case of a system failure, there is no safe switch or any means to access the door and it will remain permanently locked. The owner has to physically come and reset the system.

6 Conclusion

Our team had set out to learn and incorporate artificial intelligence into a security system, and we built a device that can detect faces and voices of users and open the object the device is protecting and alert the user (owner) of the guest user who is requesting access to the protected object. This device can be implemented on a door to protect the house, room, and other areas. It can even be used to protect safes and lockers. The application of this device is far-reaching and we made sure that regardless of the object the device is protecting, the AI powered system would be capable of providing the best security and protection to the owner of the safe. There is a lot of scope for improvements and we would love to work on this more. We as students, were able to learn a lot in the domain of AI and Machine Learning while doing this project and we hope that this device will be useful for everyone.

7 Bibliography

1 - D. Deshmukh, A., et al. "Face Recognition Using OpenCv Based On IoT for Smart Door." SSRN Electronic Journal, 2019. Crossref, doi:10.2139/ssrn.3356332.

2 - Deshwal, Amit, et al. "Smart Door Access Using Facial Recognition." International Journal of Trend in Scientific Research and Development, vol. Volume-3, no. Issue-2, 2019, pp. 442–43. Crossref, doi:10.31142/ijtsrd21363.

3 - Radzi, Syafeeza Ahmad, et al. “IoT Based Facial Recognition Door Access Control Home Security System Using Raspberry Pi.” International Journal of Power Electronics and Drive Systems (IJPEDS), vol. 11, no. 1, 2020, p. 417. Crossref, doi:10.11591/ijpeds.v11.i1.pp417-424.

4 - Prof.M.R.Sanghavi. “SMART DOOR UNLOCK SYSTEM USING FACE RECOGNITION AND VOICE COMMANDS.” International Research Journal of Engineering and Technology(IRJET), vol. 07, no. 06|, 2020, pp. 3304–07. IRJET, www.irjet.net/archives/V7/i6/IRJET-V7I6617.pdf.

5 - Tian, Xuehong. “Face Recognition System and It’s Application.” 2009 First International Conference on Information Science and Engineering, 2009. Crossref, doi:10.1109/icise.2009.583.

6 - Aditya, Eka Wahyu, et al. “Face Recognition Implementation System As A Media Access To Restricted Room With Histogram Equalization And Fisherface Methods.” 2019 International Symposium on Electronics and Smart Devices (ISESD), 2019. Crossref, doi:10.1109/isesd.2019.8909665.

7 Dwivedi, Divyansh. “Face Recognition for Beginners - Towards Data Science.” Medium, 27 Mar. 2019, towardsdatascience.com/face-recognition-for-beginners-a7a9bd5eb5c2.

8 “ASL Alphabet.” Kaggle, 22 Apr. 2018, www.kaggle.com/grassknoted/asl-alphabet.

9 Sharma, Ashish, et al. “Hand Gesture Recognition Using Image Processing and Feature Extraction Techniques.” Procedia Computer Science, vol. 173, 2020, pp. 181–190., doi:10.1016/j.procs.2020.06.022.

10 Ganapathiraju, Aravind. “Implementation of Viterbi Search Algorithm.” Implementation of Viterbi Search Algorithm.

11 “Face_recognition Package — Face Recognition 1.4.0 Documentation.” Face Recognition Documentation, face-recognition.readthedocs.io/en/latest/face_recognition.html. Accessed 9 Apr. 2021.