

**ENHANCED SECURITY USING NEURAL NETWORK
FOR PRIVACY AND SECURITY THREAT**

Submitted in partial fulfillment of the requirements for the
award of Bachelor of Technology degree in
Information Technology

By

Kaveyha G (Reg.No.39120052)

Kalari Likitha (Reg.No.39120061)



**DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI - 600119**

MAY - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

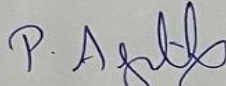
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

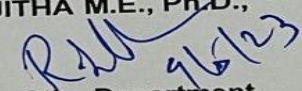
DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Kaveyha G (Reg. No. 39120052)**, and **Kalari Likitha (Reg. No. 39120061)** who carried out the Project Phase-2 entitled "**ENHANCED SECURITY USING NEURAL NETWORK FOR PRIVACY AND SECURITY THREAT**" under my supervision from January 2023 to May 2023.

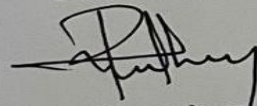

Internal Guide

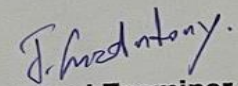
Dr P. AJITHA M.E., Ph.D.,


Head of the Department

Dr. R. SUBHASHINI M.E., Ph.D.,

Submitted for Viva voce Examination held on 10-05-2023


Internal Examiner


External Examiner

DECLARATION

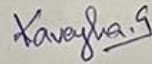
I Kaveyha G (Reg. No 39120052) here by declare that the Project Phase-2 Report entitled "ENHANCED SECURITY USING NEURAL NETWORK FOR PRIVACY AND SECURITY THREAT" done by me under the guidance of Dr. P. AJITHA M.E., Ph.D., is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 10/05/2023

Kaveyha G

PLACE: Chennai

SIGNATURE OF THE CANDIDATE



ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. R. SUBHASHINI M.E., Ph.D.**, Head of the Department, Department of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. P. AJITHA M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

ABSTRACT

In the modern world, the IoT is at its peak. The world is becoming smarter, home automation is emerging. Smart Door control system is the latest technology in home automation. The main purpose of smart Door technology is to provide complete security to the door, ease and comfort for users. The aim of this project is to enlarge the door automation technique using Embedded and android phone. The PC capturing an image of a person as soon as the bell is being pressed and sends the still image to the android device through telegram app and alerts the owner about the person standing at the door. The entire systems programming has been established in python and the Embedded c for a Controller. The world is at the brink of a new digital revolution and Internet of Things (IoT)-based Cyber Physical security architecture” and “Internet of Things” both of which are listed on Gartner’s 2016 top 10 strategic technology trends. In addition to the services offered by other instantiations of IoT, smart vehicular networks transform vehicles into formidable sense and-move platforms and assist in safe navigation, pollution control, reducing land used for parking, energy efficiency, lower delay, passenger comfort, and congestion management through smooth traffic flow. In last decade, when advances in deep learning, computing systems for route planning, communication standards, and image processing made it a reality. The main benefits of autonomous vehicles include reducing mobility costs by sharing cars, simplifying the introduction of alternative fuels, reducing traffic accidents and fatalities, better road utilization (fewer roads will be needed), reducing climate change (higher passengers per mile per gallon), allowing elderly and physically challenged people to live more independently and improve disaster response among other things. 18% of crashes with injuries are attributable to distractions caused by smartphones, an issue that is more prevalent with younger generations. This project focuses on communication and security concerns associated with IOT. This project also demonstrates the feasibility of cyber threats to such archetypes. Now a days there is a growing interest in internet of things for security systems.

TITLE OF CONTENTS

Chapter No.	TITLE	Page No.
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	
	1.1 Overview	9
	1.2 Problem Statement	10
2	LITERATURE SURVEY	
	2.1 Inferences from Literature Survey	12
	2.2 Open problems in Existing System	12
3	REQUIREMENTS ANALYSIS	
	3.1 Feasibility Studies/Risk Analysis of the Project	13
	3.2 Software Requirements Specification Document	13
4	DESCRIPTION OF THE PROPOSED SYSTEM	
	4.1 Project Management Plan	22
	4.2 System Architecture	23
	4.3 Selected Methodology	23
	4.4 Description of Software for Implementation	25
	4.5 Transition/Software to operation plan	26
5	DESIGN & DEVELOPMENT	
	5.1 Block Diagram of Hardware Components	27
	5.2 L298 N motor driver	27
	5.3 Arduino UNO	28
	5.4 9V Battery	29
	5.5 Liquid crystal display	29
	5.6 USB Cable	30
	5.7 Push Button	31
	5.8 DVD Loader	31

6	RESULT & DISCUSSION	
	6.1 Result	32
7	CONCLUSION & FUTURE WORK	
	7.1 Conclusion	34
	7.2 Future Work	34
	7.3 Research Issues	34
	7.4 Implementation Issues	35
	REFERENCES	60
	APPENDIX	
	A. SOURCE CODE	36
	B. SCREEN SHOT	50
	C. RESEARCH PAPER	54

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	Page No.
4.2.1	System Architecture	22
4.3.1	Haar Cascade	23
4.3.2	Feature Extraction	24
5.1.1	Block Diagram of Hardware Components	27
5.2.1	L298 N motor driver	28
5.3.1	Arduino UNO	28
5.4.1	9V Battery	29
5.5.1	Liquid Crystal Liquid	30
5.6.1	USB Cable	30
5.7.1	Push Button	31
5.8.1	DVD loader	31
6.1.1	User Known	32
6.1.2	User Unknown	32
6.1.3	Distance Accuracy	33
6.1.4	Distance Light Condition	33

CHAPTER 1

INTRODUCTION

The world is at the brink of a new digital revolution and Internet of Things (IoT)-based Cyber Physical security architecture” and “Internet of Things” both of which are listed on Gartner’s 2016 top 10 strategic technology trends. In addition to the services offered by other instantiations of IoT, smart vehicular networks transform vehicles into formidable sense and-move platforms and assist in safe navigation, pollution control, reducing land used for parking, energy efficiency, lower delay, passenger comfort, and congestion management through smooth traffic flow.

The idea of autonomous vehicles dates back to 20th century. but it wasn’t until last decade when advances in deep learning, computing systems for route planning, communication standards, and image processing made it a reality. The main benefits of autonomous vehicles include reducing mobility costs by sharing cars, simplifying the introduction of alternative fuels, reducing traffic accidents and fatalities, better road utilization (fewer roads will be needed), reducing climate change (higher passengers per mile per gallon), allowing elderly and physically challenged people to live more independently and improve disaster response among other things. 18% of crashes with injuries are attributable to distractions caused by smartphones, an issue that is more prevalent with younger generations.

This project focuses on communication and security concerns associates with IOT. This project also demonstrates the feasibility of cyber threats to such archetypes. Now a days there is a growing interest in interest of things for security systems.

1.1 OVERVIEW

The major goal of this system is to record a picture, video in real-time and determine if a human face is there using the Computer Vision approach. After identifying human faces in a picture and analysing, if the face is recognised, if the face is not recognized then the footage will be captured and will be send to Telegram mobile application.

1.2 PROBLEM STATEMENT

The solution is based on raspberry pi and other Micro-controllers which works out on a higher side and there is no integration of real time footage sharing in the existing system. Main intention is to provide the user with live footage-based approval system connected through IoT and Telegram Chatbot. This Concept works in a way, when button is connected to the microcontroller of the project, when the button is pressed the camera gets activated and starts detecting the face using Haar cascade and LBPH algorithm and the result is transmitted through serial communication which is connected to the microcontroller through pc. When the face matches, python will write the data to the Arduino microcontroller and that in turn will open the door automatically if the face is recognized and if the face is not recognized then the footage will be captured and will be sent to Telegram mobile Application.

CHAPTER 2

LITERATURE SURVEY

By Hteik Htar Lwin, Aung Soe Khaing, Hla Myo Tun [1] “Automatic Door Access System Using Face Recognition”. In this system, automatic face recognition and detection is done by MATLAB program on PC. Microcontroller is used to control the door access system depending on incoming data sent from PC. Door is opened immediately after confirming that the person is authenticated. After 2 seconds, door is closed automatically. However, 2 seconds are not enough time for a person to enter. So, longer time should be set for real-time conditions. Viola Jones algorithm is used for face detection. Since, this algorithm can only detect frontal view of the face correctly, this system has limitations.

By Sandesh Kulkarni, Minakshee Bagul, Akansha Dukare, [2] “Face Recognition System Using IoT”. The system developed was cost and power efficient. In this system, Viola Jones Algorithm is used for face detection that eliminates face candidates using a cascade of stages but the structural information captured by it is limited.

In Paul Viola [3] an efficient face detection algorithm. Introduces concept of integral image, Adaboost classifier resulting in efficient and fast face detection

Viola Jones [4] understood that LBPH (Local Binary Pattern Histogram) is better when it comes to discriminative power. Hence, a person can be identified not only from frontal view but also from different pose and illumination.

By Yichun Shi and Anil K. Jain [5]. Current embedding methods used for face recognition are able to achieve high performance in controlled settings. These methods work by taking an image of a face and storing data about that face in a latent semantic space. However, when tested in fully uncontrolled settings, the current methods cannot perform as well. An example of such a case would be face recognition in surveillance videos, where the quality of the video may be low. To help address this issue, the authors propose Probabilistic Face Embeddings (PFEs). The authors propose a method for converting existing deterministic embeddings into

PFEs. Most importantly, the authors state that this method effectively improves performance in face recognition models.

By Shifeng Zhang, Xiaobo Wang, Hailin Shi [6]. In order to support face anti-spoofing research, the authors of this paper introduce a multi-modal face anti-spoofing dataset named CASIASURF. As of the writing of this paper, it is the largest open dataset for face anti-spoofing. Specifically, the dataset includes 21,000 videos taken of 1,000 subjects in RGB, Depth, and IR modalities. In addition to the dataset, the authors present a novel multi-modal fusion model as a baseline for face anti-spoofing.

By Fei Wang, Liren Chen [7]. The University of California San Diego, and Nanyang Technological University studied the effects of noise in large-scale face image datasets. Many large datasets are prone to label noise, due to their scale and cost-effective nature. With this paper, the authors aim to provide knowledge around the source of label noise and the consequences it has in face recognition models. Additionally, they aim to build and release a clean face recognition dataset titled IMDB-Face. Two of the main goals of the study were to discover the effects of noise on final performance, and determine the best strategy to annotate face identities.

2.1 INFERENCES FROM LITERATURE SURVEY

One thing is for sure that any day security plays a major role in day today life based on the inferences from the literature review, an innovative solution which reduces the risk and provides effective solution to overcome the existing system with affordable cost.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

The Open Problem in the Existing System are almost all the solution are based on raspberry pi and other Micro-controllers which works out on a higher side and there is no integration of real time footage sharing in the existing system the intention is to provide the user with live footage-based approval system connected through IoT and Telegram Chatbot.

CHAPTER-3

REQUIREMENT ANALYSIS

3.1 RISK ANALYSIS OF THE PROJECT

In the Existing System, Complex Tasks are carried by Microcontroller and cost becomes high for the user and it also not possible to carry and set it every time. This solution is to overcome the problem which has capability to detect Live Facial Recognition and Integrate the Footage to the telegram chatbot and based on the authorization of the user, they can authenticate that person to enter.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

- **Programming:** The way in which the software is created today has changed dramatically the last 30 years, from the childhood of personal computers in the early 80s to today's powerful devices such as Smartphones, Tablets and PCs. The Internet has also changed the way of usage in devices and software. Some still have traditional desktop applications, but Web Sites, Web Applications and so- called Apps for Smartphones, etc. are dominating the software market today.
- **The need to find and learn Programming Languages** that are suitable for the New Age of Programming. Today several thousand different Programming Languages are being in use.
- **Software Engineering:** Software Engineering is the discipline for creating software applications. A systematic approach to the design, development, testing, and maintenance of software. The main parts or phases in the Software Engineering process are:
 - Planning
 - Requirements Analysis
 - Design
 - Implementation
 - Testing

➤ Deployment and Maintenance

Python is a fairly old Programming Language (1991) compared to many other Programming Languages like C# (2000), Swift (2014), Java (1995), PHP (1995).

- Python is highly extendable due to its high number of free available Python Packaged and Libraries
- Python can be used on all platforms (Windows, macOS and Linux).
- Python is multi-purpose and can be used for to program Web Applications, Enterprise Applications and Embedded Applications, and within Data.
- Science and Engineering Applications.
- The popularity of Python is growing fast.
- Python is open source and free to use

The growing Python community makes it easy to Find documentation, code examples and get help when needed

- Introduction to Python: Python is an open source and cross-platform programming language, that has become increasingly popular over the last ten years. It was First released in 1991. Latest version is 3.7.0. CPython is the reference implementation of the Python programming language. Written in C, CPython is the default and most widely-used implementation of the language.
- Python is a multi-purpose programming language (due to its many extensions), examples are scientific computing and calculations, simulations, web development (using, e.g., the Django Web framework), etc. Python is an object-oriented programming language (OOP), but you can use Python in basic application without the need to know about or use the object- oriented features in Python
- Arduino: Arduino Integrated Development Environment (IDE) v1 Learn how the Arduino IDE v1 works, such as compiling & uploading sketches, file management, installing dependencies and much more.

- The Arduino Integrated Development Environment – or Arduino Software (IDE) contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.
- Writing Sketches: Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension. `.ino`. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.
- NB: Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension. `.pde`. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the. `.ino` extension on save.
IDE VERIFY File: Verify Checks your code for errors compiling it.
- Upload Compiles your code and uploads it to the configured board. See uploading below for details. Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer".
- IDE NEW File: New Creates a new sketch, IDE OPEN File: Open Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content. Note: due to a bug in Java, this menu does not scroll; if you need to open a sketch late in the list, use the File Sketchbook menu instead.

- IDE SAVE File: Save Saves your sketch, IDE SERMON File: Serial Monitor Opens the serial monitor. Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.
- File New Creates a new instance of the editor, with the bare minimum structure of a sketch already in place. Open Allows to load a sketch file browsing through the computer drives and folders. Open Recent Provides a short list of the most recent sketches, ready to be opened. Sketchbook Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.
- Examples Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library. Close Closes the instance of the Arduino Software from which it is clicked.
- Save Saves the sketch with the current name. If the file has not been named before, a name will be provided in a "Save as." window. Save as... Allows to save the current sketch with a different name. Page Setup It shows the Page Setup window for printing. Print Sends the current sketch to the printer according to the settings defined in Page Setup. Preferences Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface. Quit Closes all IDE windows. The same sketches open when quit was chosen will be automatically reopened the next time you start the IDE.
- Edit Undo/Redo Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo. Cut Removes the selected text from the editor and places it into the clipboard.

- **Copy** Duplicates the selected text in the editor and places it into the clipboard. **Copy for Forum** Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.
- **Copy as HTML** Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages. **Paste** Puts the contents of the clipboard at the cursor position, in the editor. **Select All** Selects and highlights the whole content of the editor. **Comment/Uncomment** Puts or removes the // comment marker at the beginning of each selected line. **Increase/Decrease Indent** Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.
- **Find** Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options. **Find Next** Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position. **Find Previous** Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position. **Sketch Verify/Compile** Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area. **Upload** Compiles and loads the binary file onto the configured board through the configured Port. **Upload Using Programmer** This will overwrite the bootloader on the board; you will need to use **Tools > Burn Bootloader** to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a **Tools -> Burn Bootloader** command must be executed.
- **Export Compiled Binary** Saves a .hex file that may be kept as archive or sent to the board using other tools. **Show Sketch Folder** Opens the current sketch folder. **Include Library** Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files. **Add File...** Adds a supplemental file to the

sketch (it will be copied from its current location). The file is saved to the data subfolder of the sketch, which is intended for assets such as documentation. The contents of the data folder are not compiled, so they do not become part of the sketch program.

- **Tools: Auto Format** This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more. **Archive Sketch** Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch. **Fix Encoding & Reload** Fixes possible discrepancies between the editor char map encoding and other operating systems char maps. **Serial Monitor** Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.
- **Board** Select the board that you are using. See below for descriptions of the various boards. **Port** This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu. **Programmer** For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you will not need this, but if you're burning a bootloader to a new microcontroller, you will use this.
- **Burn Bootloader** The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino board but is useful if you purchase a new ATmega microcontroller (which normally come without a bootloader).
- **Ensure that you've selected the correct board from the Boards menu before burning the bootloader on the target board.** This command also set the right fuses.
- **Help:** Here you find easy access to a few documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this

guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website. Find in Reference This is the only interactive function of the Help menu: it directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

- Sketchbook: The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog. Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino. Tabs, Multiple Files, and Compilation
- Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h). Before compiling the sketch, all the normal Arduino code files of the sketch (.ino, .pde) are concatenated into a single file following the order the tabs are shown in. The other file types are left as is.
- Uploading: Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an UNO or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the sports section of the Windows Device Manager. On Linux, it should be

`/dev/ttyACMx` , `/dev/ttyUSBx` or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

- When you upload a sketch, you are using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e., when the board resets).
- Libraries: Libraries provide extra functionality for use in sketches, e.g., working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more `#include` statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its `#include` statements from the top of your code.
- There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.
- Third-Party Hardware: Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries,

bootloaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or you will override the built-in Arduino platform.) To uninstall, simply delete its directory.

- For details on creating packages for third-party hardware, see the Arduino Platform specification. Serial Monitor: This displays serial sent from the Arduino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to `Serial.begin` in your sketch. Note that on Windows, Mac, or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor. Please note that the Serial Monitor does not process control characters; if your sketch needs a complete management of the serial communication with control characters, you can use an external terminal program and connect it to the COM port assigned to your Arduino board.
- You can also talk to the board from Processing, Flash, MaxMSP, etc (see the interfacing page for details).

CHAPTER 4

DESCRIPTION OF THE PROPOSED SYSTEM

Security and safety are becoming more and more popular day by day. Hence, it is getting improved and used in daily life. Now a days, technology has become an integrated part of people's lives therefore the security of one's home, office or organization must not be left behind. The goal is to design a system with minimum human intervention. Smart Receptionist with smart lock system is mainly designed and developed for security and safety purposes.

This smart security system is used to see a visitor when the main door of the office or organization is closed. The purpose of this system is to control the door lock and authorization using Telegram Application. In this system, whenever a person rings the bell, image of the person is captured by the camera. If person's image is present in database, he/she will be allowed in else if image is not present in the database, then the smart lock system will send the image to Telegram App for Authorization. Once the image is sent to the Telegram Application, option will be available with user of either locking or unlocking the door. If the user clicks on unlock button the door will open and person standing on the door will be allowed in and if the user clicks on lock button the door will remain locked and person will not be allowed in door.

4.1 PROJECT MANAGEMENT PLAN

1: Hardware Integration:

Hardware integration is also called as System integration or Hardware-Software integration. Which requires software to connect with one or more pieces of physical hardware to make a single usable system.

2: Python Programming for Facial Recognition with Integration with Telegram Chat-bot:

Python programming is used for facial detection. The major part of the project is to detect faces. When the camera gets activated, a message will be updated in the telegram chat-bot. By this action the door can be easily monitored.

3: Embedded 'C' Programming & Testing:

Here embedded C programming is being used for the hardware. After successful connections of the hardware modules, it can further proceed for testing.

4.2 SYSTEM ARCHITECTURE

This is the system architecture of the project. This represents the setp by step procedure of the implementation. The procedure starts with Face acquisition and ends if two results if face matches it show matching if not matches it shows not matching.

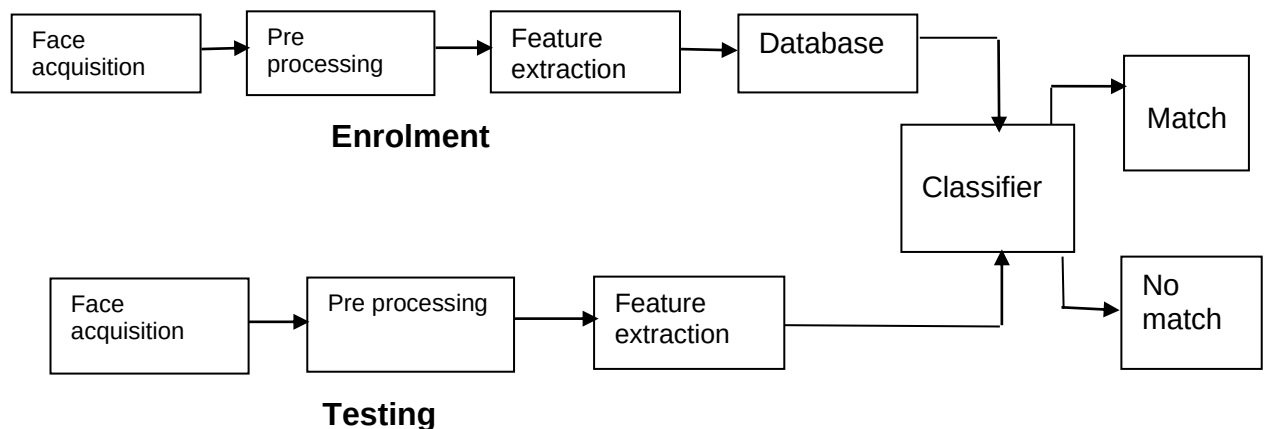


Fig 4.2.1 System Architecture

4.3 SELECTED METHODOLOGY

Here the selected Haar Cascade and Linear Binary Pattern (LPBH) Algorithm is used. This algorithms is used for easy detection of faces in camera.

Haar Cascades

A sequence of rescaled “square-shaped” functions which together form a wavelet family or basis. It is based on the Haar Wavelet technique to analyze pixels in the image into squares by function.

This uses machine learning techniques to get a high degree of accuracy from what is called “training data”. This uses “integral image” concepts to compute the “features” detected. Haar Cascades use the Ada boost learning algorithm which selects a small number of important features from a large set to give an efficient result of classifiers.

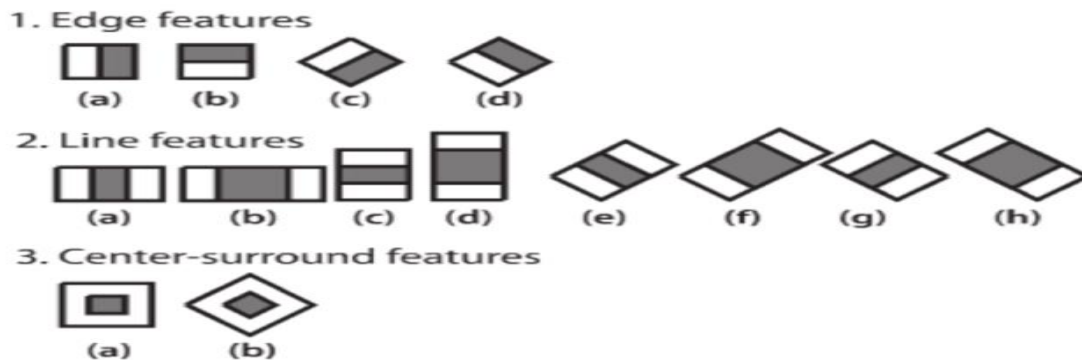


Fig: 4.3.1 Haar Cascades

- **Face Detection** determines the location and sizes of human faces in arbitrary (digital) images.
- In **Face Recognition**, the use of Face Detection comes first to determine and isolate a face before it can be recognized.

Feature Extraction

Haar Cascades use machine learning techniques in which a function is trained from a lot of positive and negative images. This process is feature extraction.

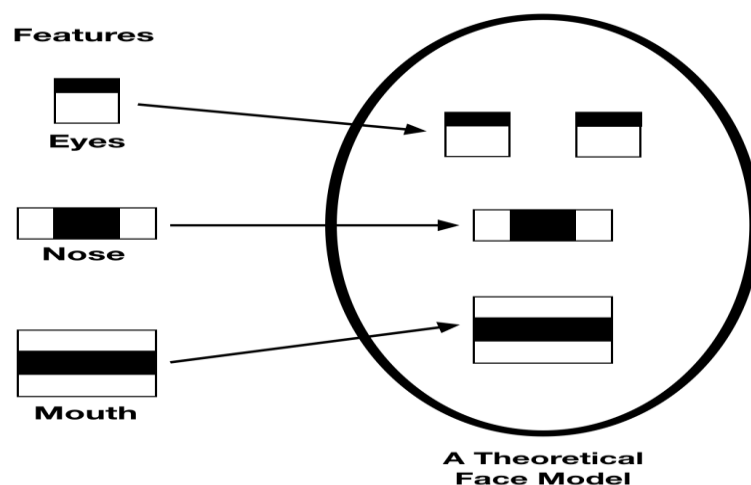


Fig: 4.3.2 Feature Extraction

- In feature extraction, the algorithm uses training data to best identify features that it can consider a face.
- The training data used in this project is an XML file called: `haarcascade_frontalface_default.xml`.

4.4 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION

This Project program is developed using Python where facial recognition plays a major role and connected by a powerful machine learning tool with the hardware (Embedded Systems) to make in more efficient, effective and to provide real time solution. The Concept work in the way in which a button connected to the microcontroller of the project once when the button is pressed the camera gets activated and starts detecting the face using Haar cascade and LBPH algorithm and the result is transmitted through serial communication which is connected to the microcontroller through pc once the face matches python will write the data to the Arduino microcontroller and that in turn will open the door automatically if the face is recognized and if the face is not recognized then the footage will be captured and will be sent to Telegram mobile Application through Bot Father once that process is completed the user now has the authority either to authenticate or not to authenticate that person if authentication is successful then the door opens else the door remains closed.

This project, implies a system for locating faces in digital images. These are in JPEG format only. The face recognition and face detection are not the same, but one depends on the other. In this case face recognition needs face detection for making an identification to “recognize” a face.

Face detection uses *classifiers*, which are algorithms that detects what is either a face (1) or not a face (0) in an image. Classifiers have been trained to detect faces using thousands to millions of images in order to get more accuracy. OpenCV uses two types of classifiers, LBP (Local Binary Pattern) and Haar Cascades. This will be using the latter classifier.

A sequence of rescaled “square-shaped” functions which together form a wavelet family or basis. It is based on the Haar Wavelet technique to analyze pixels in the image into squares by function. This uses machine learning techniques to get a high degree of accuracy from what is called “training data”. This uses “integral image” concepts to compute the “features” detected. Haar Cascades use the Adaboost learning algorithm which selects a small number of important features from a large set to give an efficient result of classifiers.

4.5 TRANSITION/ SOFTWARE TO OPERATIONS PLAN

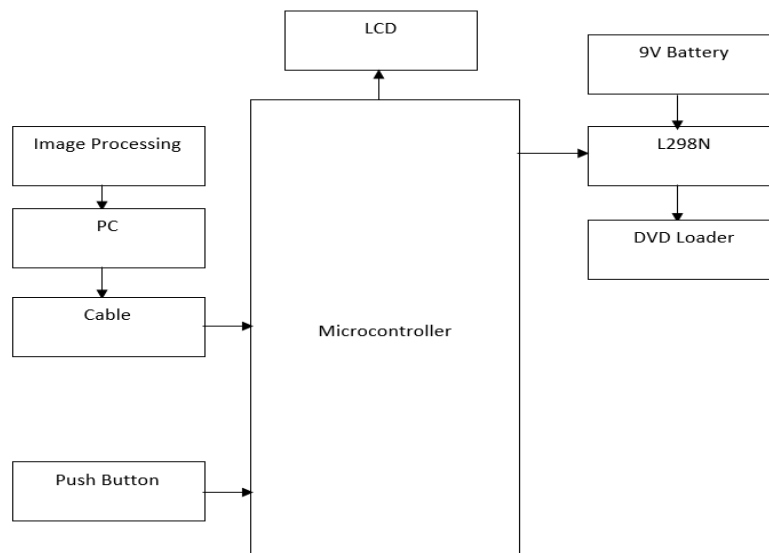
Main Objective of the Project is to Build a Robust IoT Based Security Solution which could be used in multiple Scenarios like for Class Rooms During Pandemic Situations, Bank Locker Rooms, Women’s Hostel, Confidential Meeting Rooms to authenticate only authorized person by Using Embedded ‘c’ & Python Language Integrated with Telegram Chat-Bot.

CHAPTER 5

DESIGN AND DEVELOPMENT

5.1 BLOCK DIAGRAM OF HARDWARE COMPONENTS

This is the block diagram of Hardware components which represents the connection between the components and what components are used in the project.



5.1.1 Block Diagram of Hardware Components

5.2 L298 N MOTOR DRIVER

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A. This depends on the voltage used at the motors VCC. The module has an onboard 5V regulator which is either enabled or disabled using a jumper. The 5V pin can be used as output, for example for powering the Arduino board.

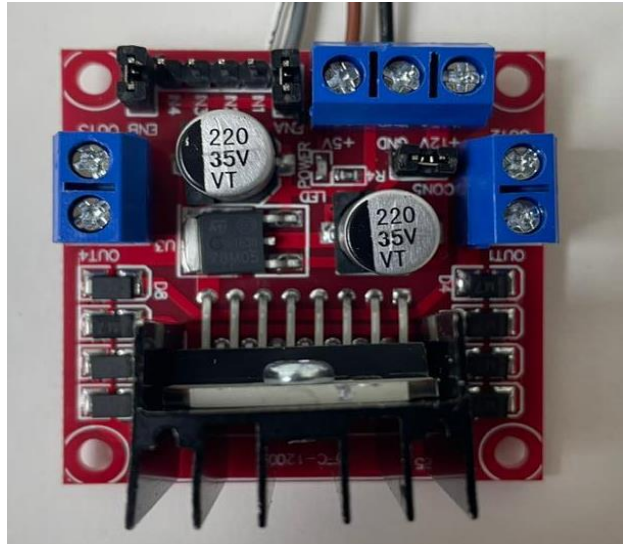


fig 5.2.1 L298 n motor driver

5.3 ARDUINO UNO

Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

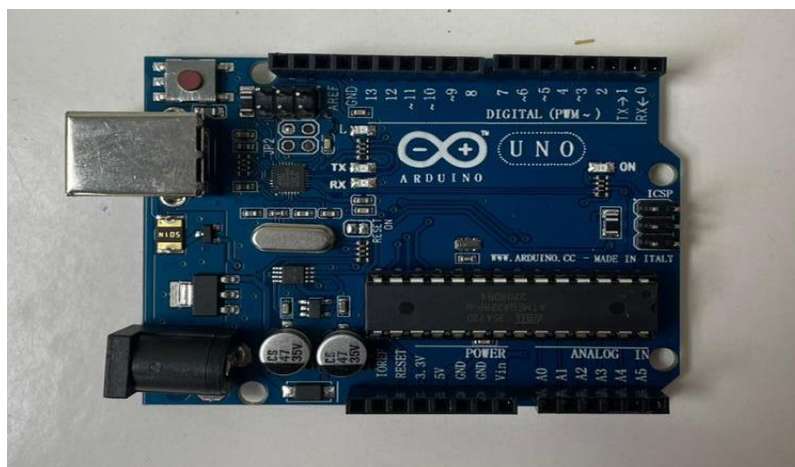


Fig 5.3.1 Arduino UNO

5.4 9V Battery

The nine-volt battery, or 9-volt battery, is an electric battery that supplies a nominal voltage of 9 volts. Actual voltage measures 7.2 to 9.6 volts, depending on battery chemistry. Batteries of various sizes and capacities are manufactured; a very common size is known as PP3, introduced for early transistor radios.



Fig 5.4.1 9V Battery

5.5 LIQUID CRYSTAL DISPLAYS

Liquid crystal displays (LCDs) have materials which combine the properties of both liquids and crystals. Rather than having a melting point, they have a temperature range within which the molecules are almost as mobile as they would be in a liquid, but are grouped together in an ordered form similar to a crystal. An LCD consists of two glass panels, with the liquid crystal material sandwiched in between them. The inner surface of the glass plates is coated with transparent electrodes which define the character, symbols or patterns to be displayed. Polymeric layers are present in between the electrodes and the liquid crystal, which makes the liquid crystal molecules to maintain a defined orientation angle.

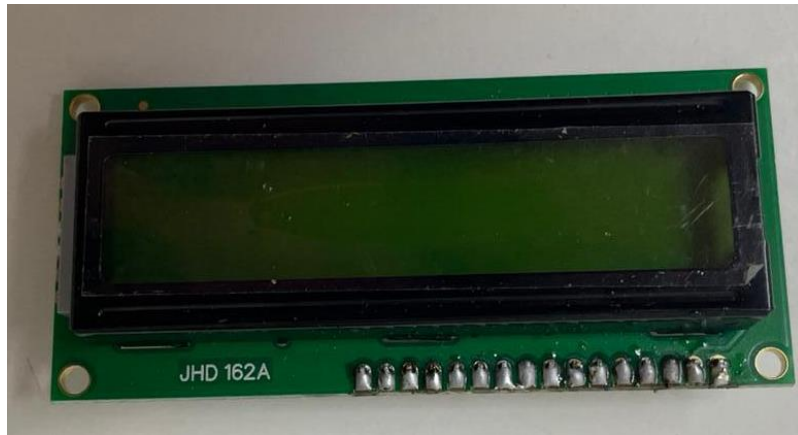


Fig 5.5.1 Liquid Crystal Displays

5.6 USB Cable

Use it to connect Arduino Uno, Arduino Mega 2560, Arduino 101 or any board with the USB female A port of your computer. Cable length is approximately 100cm. Cable color and shape may vary slightly from image as our stock rotates. USB cable type A/B Standard USB 2.0 cable.



Fig 5.6.1 USB Cable

5.7 Push Button

Pushbuttons or switches connect two points in a circuit when you press them. This example turns on the built-in LED pin 13 when you press the button.

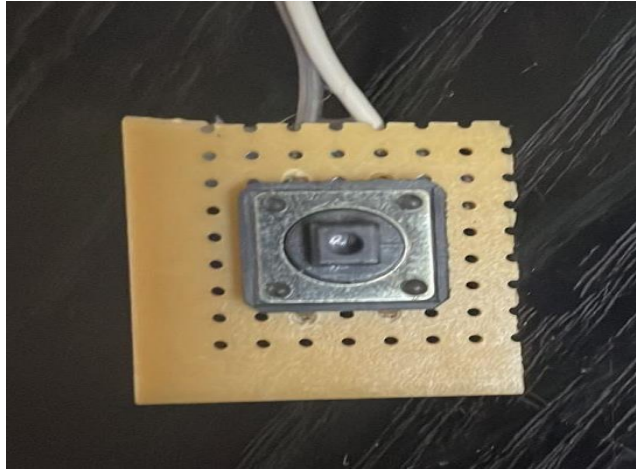


Fig 5.7.1 Push Button

5.8 DVD Loader

This is used as a sample door for this project. The automatic opening and closing of the DVD loader symbolizes the automatic moment of the door. As the instructions given the DVD loader acts according to the instructions given.



Fig 5.8.1 DVD Loader

CHAPTER 6

RESULTS & DISCUSSION

6.1 RESULTS

The primary goal of the project is to develop a reliable IOT-based security solution that can be used in a variety of scenarios, such as in bank locker rooms, women's hostels, and confidential meeting rooms, to authenticate only authorized people using embedded 'c' and Python language integrated with Telegram Chat-Bot.

```
import cv2

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.yml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);

font = cv2.FONT_HERSHEY_SIMPLEX

#initiate id counter
id = 0

# names related to ids: example ==> id=1, etc
names = ['none', 'id 1', 'id 2']

# Initialize and start realtime video capture
cam = cv2.VideoCapture(0)
#cam.set(3, 480) # set video width
#cam.set(4, 480) # set video height

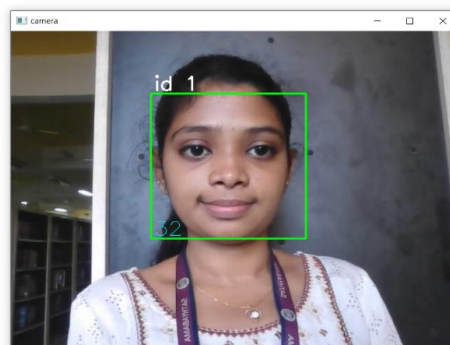
while True:
    ret, img = cam.read()
    img = cv2.flip(img, 1) # Flip vertically
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.1,
        minNeighbors = 5,
        minSize = (30, 30),
    )

    for (x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)

        id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
        conf=round(100 - confidence)

        # Check if confidence is less than 100 ==> "0" is perfect match
        if ( conf > 30):
            id = names[id]
            confidence = " {0}%".format(conf)
```



6.1.1 USER KNOWN

```
import cv2

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.yml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);

font = cv2.FONT_HERSHEY_SIMPLEX

#initiate id counter
id = 0

# names related to ids: example ==> id=1, etc
names = ['none', 'id 1', 'id 2']

# Initialize and start realtime video capture
cam = cv2.VideoCapture(0)
#cam.set(3, 480) # set video width
#cam.set(4, 480) # set video height

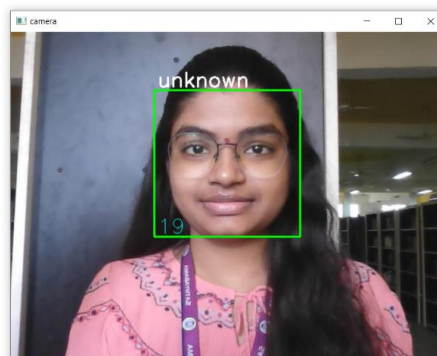
while True:
    ret, img = cam.read()
    img = cv2.flip(img, 1) # Flip vertically
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.1,
        minNeighbors = 5,
        minSize = (30, 30),
    )

    for (x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)

        id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
        conf=round(100 - confidence)

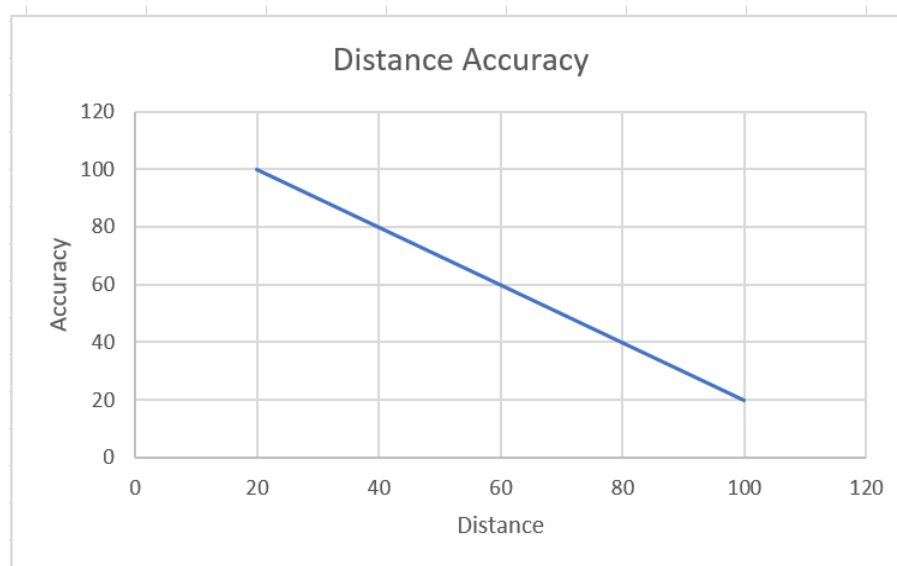
        # Check if confidence is less than 100 ==> "0" is perfect match
        if ( conf > 30):
            id = names[id]
            confidence = " {0}%".format(conf)
```



6.1.2 USER UNKNOWN

Distance-Accuracy

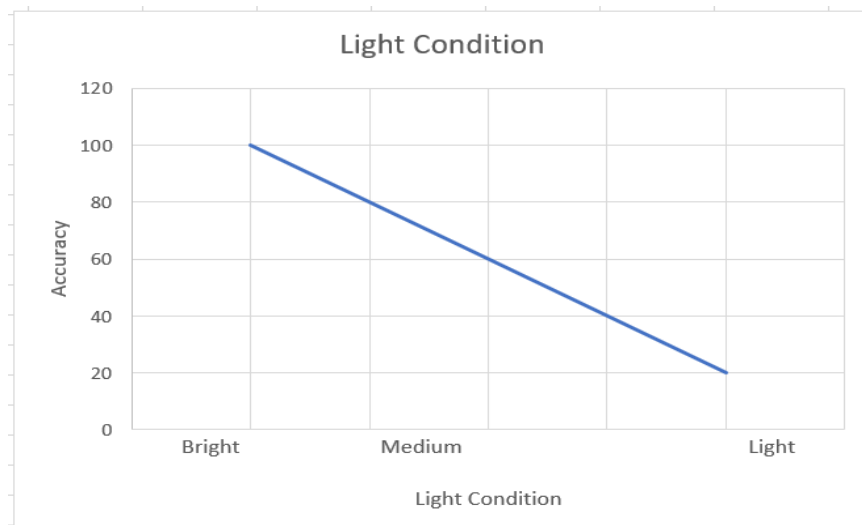
Following graph is the result of the accuracy of project, which determines the variations of Distance accuracy. As the distance increases the accuracy level may decrease.



6.1.3 DISTANCE ACCURACY

Distance-Light Condition

Following graph is the result of the accuracy of project, which determines the variations of Distance Light condition. As the light condition decreases the accuracy level may decrease.



6.1.4 DISTANCE LIGHT CONDITION

CHAPTER 7

CONCLUSION & FUTURE WORK

7.1 CONCLUSION

By this enhanced security project, it is concluded that being busy in the day to day lives it can't be physically possible to open the door if someone comes to the house. So, by this Automatic door operation, being anywhere any person can monitor their house and also allow the people who come to their house.

Automatic door lock systems offer a range of benefits compared to traditional lock systems, such as convenience, security, and flexibility. With features such as remote access, real-time notifications, user management and allow for greater control and monitoring of access the house or business. As a result, the house can be monitored without any fear.

7.2 FUTURE WORK

This smart door locking and unlocking system is of wire based. This can be further implemented into wireless by using raspberry-pi. The codes and the connections are stored in raspberry-pi.

7.3 RESEARCH ISSUES

During the implementation of this project, some issues have to be addressed to ensure the success of the project.

Information or reference papers regarding the subject were not widely available. In preview of that it was also found that camera device should be connected and the door should be physically opened. But here there has been a new thought included where the door opens automatically. Based on the aforementioned information gathered it was all about hardware & software separately and joined both of them to get our project implemented.

7.4 IMPLEMENTATION ISSUES

During the implementation of this project, a few persistent issues were discovered, which are,

One of the main implementation issues was the need for good network connectivity. To establish communication between the Arduino UNO board to the telegram chat-bot network connection is mandatory. The second issue to be addressed is the need for standard light condition, a stable light condition was necessary for capturing and recognizing the faces.

APPENDIX

A. SOURCE CODE

CAMERA

```
import cv2
cap = cv2.VideoCapture(0)
while (1):
    #Read the frame
    ret, frame = cap. Read ()
    if ret==True:
        #frame = cv2.flip(frame,1)
        #Show the frame
        cv2.imshow('frame', frame)
        k = cv2.waitKey(1) & 0xff # Press 'ESC' for exiting video
        if k == 27:
            break
    else:
        break
# Release everything if job is finished
cap. release ()
cv2.destroyAllWindows()
```

IMAGE SPLIT:

```
import cv2
image = cv2.imread('rgb.png')
B, G, R = cv2.split(image)
# Corresponding channels are separated
cv2.imshow("original", image)
cv2.waitKey(0)
cv2.imshow("blue", B)
cv2.waitKey(0)
cv2.imshow("Green", G)
cv2.waitKey(0)
cv2.imshow("red", R)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

FACE

```
import cv2
import os
cam = cv2.VideoCapture(0)
cam.set(3, 640) # set video width
cam.set(4, 480) # set video height
face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
print ("\n [INFO] Initializing face detection. Look the camera and wait ...")
while (True):
    ret, img = cam.read ()
    img = cv2.flip(img, 1) # flip video image vertically
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_detector.detectMultiscale (gray, 1.3, 5)
    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x+w, y+h), (255,0,0), 2)
        cv2.imshow('image', img)
    k = cv2.waitKey(1) & 0xff # Press 'ESC' for exiting video
    if k == 27:
        break
# Do a bit of cleanup
cam.release ()
cv2.destroyAllWindows()
```

Face Dataset:

```
import cv2
import os
cam = cv2.VideoCapture(0)
#cam.set(3, 240) # set video width #cam.set(4, 240) # set video height
face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
# For each person, enter one numeric face id
face_id = input("\n enter user id end press <return> ==> ")
print("\n [INFO] Initializing face capture. Look the camera and wait ...")
# Initialize individual sampling face count
count = 0
while(True):
    ret, img = cam.read()
    img = cv2.flip(img, 1) # flip video image vertically
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_detector.detectMultiScale(gray, 1.3, 5)
    for (x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
        count += 1
        print(count) # Save the captured image into the datasets folder
        cv2.imwrite("dataset/User." + str(face_id) + '.' + str(count) + ".jpg",
gray[y:y+h,x:x+w])
        cv2.imshow('image', img)
        cv2.imshow('live', img)
        k = cv2.waitKey(10) & 0xff # Press 'ESC' for exiting video
        if k == 27:
            break
        elif count >= 200: # Take 200 face sample and stop video
            break
    print("\n [INFO] Exiting Program and cleanup stuff")
    cam.release()
    cv2.destroyAllWindows()
```

Face recognition:

```
import cv2
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.yml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);
font = cv2.FONT_HERSHEY_SIMPLEX
#initiate id counter
id = 0
# names related to ids: example ==> id=1, etc
names = ['none', 'id 1', 'id 2']
# Initialize and start realtime video capture
cam = cv2.VideoCapture(0)
#cam.set(3, 480) # set video width
#cam.set(4, 480) # set video height
while True:
    ret, img =cam.read()
    img = cv2.flip(img, 1) # Flip vertically
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.1,
        minNeighbors = 5,
        minSize = (30, 30),
    )
    for(x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
        id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
        conf=round(100 - confidence)
        # Check if confidence is less than 100 ==> "0" is perfect match
        if (conf > 30):
```

```

        id = names[id]
        confidence = " {0} %".format(conf)

    else:
        id = "unknown"
        confidence = " {0}%".format(conf)
    cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
    cv2.putText(img, str(conf), (x+5,y+h-5), font, 1, (255,255,0), 1)
cv2.imshow('camera',img)
    k = cv2.waitKey(1) & 0xff # Press 'ESC' for exiting video
    if k == 27:
        break
# Do a bit of cleanup
print("\n [INFO] Exiting Program and cleanup stuff")
cam.release()
cv2.destroyAllWindows()

```

Face trainer:

```
import cv2
import numpy as np
from PIL import Image
import os

# Path for face image database
path = 'dataset'
recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");

# function to get the images and label data
def getImagesAndLabels(path):
    imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
    faceSamples=[]
    ids = []
    for imagePath in imagePaths:
        PIL_img = Image.open(imagePath).convert('L') # convert it to grayscale
        img_numpy = np.array(PIL_img,'uint8')
        id = int(os.path.split(imagePath)[-1].split(".")[1])
        faces = detector.detectMultiScale(img_numpy)
        #print(img_numpy)
        for (x,y,w,h) in faces:
            faceSamples.append(img_numpy[y:y+h,x:x+w])
            ids.append(id)
    return faceSamples,ids

print ("\n [INFO] Training faces. It will take a few seconds. Wait ...")
faces,ids = getImagesAndLabels(path)
#cv2.imshow('im',faces)
recognizer.train(faces, np.array(ids))
# Save the model into trainer/trainer.yml
recognizer.write('trainer/trainer.yml')
# recognizer.save() worked on Mac, but not on Pi
# Print the numer of faces trained and end program
print("\n [INFO] {0} faces trained. Exiting Program".format(len(np.unique(ids))))
```

Door Bell:

```
import time
import cv2
import datetime
import sys
import telepot
import datetime as dt
from subprocess import call
import subprocess
import os.path
import cv2
import datetime as dt
from subprocess import CalledProcessError
import serial

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.yml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);
font = cv2.FONT_HERSHEY_SIMPLEX

#iniciate id counter
id = 0
chat_id=1393309594
chat_id2=234352074
img_txt=""
names = ['none', 'Kaveyha', 'id 2']
# Initialize and start realtime video capture
cam = cv2.VideoCapture(0)
#cam.set(3, 240) # set video width
#cam.set(4, 240) # set video height
```

```

# Define min window size to be recognized as a face
arduinoData = serial.Serial('COM3', 115200)
readdat=""
def send_close():
    global DaTa_r
    DaTa_r=""
    arduinoData.write('closedoor'.encode())
def send_open():
    global DaTa_r
    DaTa_r=""
    arduinoData.write('opendoor'.encode());
def send_open_():
    global DaTa_r
    DaTa_r=""
    print('sending command')
    arduinoData.write('open_door'.encode());
DaTa_r = ""
def getSerial():
    global DaTa_r
    if(arduinoData.in_waiting >0):
        line = arduinoData.readline()
        readdat=line.decode()
        print(readdat)
        if(readdat == "Ready\r\n"):
            print("device online")
        if(readdat == "checkStat\r\n"):
            print("Start Capturing")
            DaTa_r="run"
        if(readdat == "releaseStat\r\n"):
            DaTa_r=""
        readdat=""
def handle(msg):
    #print(msg)
    chat_id = msg['chat']['id']

```

```

command = msg['text']
#print('Got command: %s' % command)
if command == '/image':
    bot.sendMessage(chat_id,text="Taking Photos")
    ret, frame = cam.read()
    if ret==True:
        frame = cv2.flip(frame,1)
        cv2.imwrite('./photo.jpg', frame)
        cv2.imshow('camera',frame)
    else:
        bot.sendMessage(chat_id,text="Captured Failed")
#cv2.destroyAllWindows()
bot.sendPhoto(chat_id=chat_id, photo=open('./photo.jpg', 'rb'))
elif command == '/video':
    bot.sendMessage(chat_id,text="Camera is starts recording")
    time.sleep(.1)
    bot.sendMessage(chat_id,text="Hold on please for 10 sec ")

capture_duration = 10 #change here for duration of capturing video

dim=(640,480)
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('video.mp4',fourcc, 20.0, (640,480))
start_time = time.time()
while( int(time.time()) - start_time) < capture_duration ):
    ret, frame = cam.read()
    if ret==True:
        frame = cv2.flip(frame,1)
        resized = cv2.resize(frame, dim, interpolation = cv2.INTER_AREA)
        out.write(resized)
        cv2.imshow('camera',frame)
    else:
        bot.sendMessage(chat_id,text="Record Failed")
        break

```

```

k = cv2.waitKey(1) & 0xff # Press 'ESC' for exiting video
if k == 27:
    break
out.release()
#cam.release()
#cv2.destroyAllWindows()
bot.sendMessage(chat_id,text="Recoding Completed")
time.sleep(.1)
bot.sendMessage(chat_id,text="Uploading video please be patient")
time.sleep(.1)
bot.sendVideo(chat_id, video=open('./video.mp4', 'rb'))
bot.sendVideo(chat_id2, video=open('./video.mp4', 'rb'))
elif command == '/opendoor':
    print("Opening Door")
    send_open()
elif command == '/closedoor':
    print("Closing Door")
    send_close()
elif (command=='hi') or (command=='Hi') or (command=='Hello') or
(command=='hello') or (command=='Hey') or (command=='hey'):
    bot.sendMessage(chat_id,text="Hello i'm your assistant, How can I Help You \n
/help")
elif (command=='bye') or (command=='Bye') or (command=='See you') or
(command=='take care') or (command=='Hey') or (command=='hey'):
    bot.sendMessage(chat_id,text=", Leave Everything to your Assistant i'll will
safegaurd your house ,Have a Good Day")
elif (command=='/-h') or (command=='/help') or (command=='/info') :
    bot.sendMessage(chat_id,text="Hello You Can Proceed with these commands
\n /opendoor \n /closedoor \n /video \n /image")
def sendvid():
    global chat_id
    bot.sendMessage(chat_id,text="Someone is trying to reach you.. \n I'm going to
Start record now")
    time.sleep(.1)

```

```

bot.sendMessage(chat_id,text="Hold on please for 10 sec ")
print('Start recording..')
capture_duration = 10 #change here for duration of capturing video
dim=(640,480)
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('video.mp4',fourcc, 20.0, (640,480))
start_time = time.time()
while( int(time.time() - start_time) < capture_duration ):
    ret, frame = cam.read()
    if ret==True:
        frame = cv2.flip(frame,1)
        resized = cv2.resize(frame, dim, interpolation = cv2.INTER_AREA)
        out.write(resized)
        cv2.imshow('camera',frame)
    else:
        bot.sendMessage(chat_id,text="Record Failed")
        break
    k = cv2.waitKey(1) & 0xff # Press 'ESC' for exiting video
    if k == 27:
        break
out.release()
#cv2.destroyAllWindows()
#cam.release()
print('Stop recording..')
bot.sendMessage(chat_id,text="Recoding Completed")
time.sleep(.1)
bot.sendMessage(chat_id,text="Uploading video please be patient")
time.sleep(.1)
bot.sendVideo(chat_id, video=open('./video.mp4', 'rb'))
bot.sendMessage(chat_id,text="/opendoor \n /closedoor \n /video \n /image")
bot = telepot.Bot('6092521112:AAGdpT5MMrtnsYUcHLGaDSN2M31dZu_2NWQ')
bot.message_loop(handle)
#bot.sendMessage(234352074,text="Hello i'm your assistant, How can I Help You")
print('Hello Good to See you Again')

```

```

bot.sendMessage(chat_id,text="Hello i'm your assistant, How can I Help You")
bot.sendMessage(chat_id2,text="Hello i'm your assistant, How can I Help You")
def getCam():
    global DaTa_r
    global img_txt
    ret, img =cam.read()
    img = cv2.flip(img, 1) # Flip vertically
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.1,
        minNeighbors = 5,
        minSize = (30, 30),
    )
    for(x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
        id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
        conf=round(100 - confidence)
        # Check if confidence is less them 100 ==> "0" is perfect match
        if DaTa_r=="run":
            if ( conf > 40):
                send_open_()
                print ("Permission Granted")
                time.sleep(2)
                img_txt = names[id]
                confidence = " {0}%".format(conf)
                DaTa_r=""
            else:
                print ("Permission Denied")
                img_txt = "unknown"
                sendvid()
                confidence = " {0}%".format(conf)
                DaTa_r=""
    cv2.putText(img, str(img_txt), (x+5,y-5), font, 1, (255,255,255), 2)

```

```
cv2.putText(img, str(conf), (x+5,y+h-5), font, 1, (255,255,0), 1)

cv2.imshow('camera',img)
while (True):
    getSerAI()
    getCam()
    k = cv2.waitKey(1) & 0xff # Press 'ESC' for exiting video
    if k == 27:
        break
cam.release()
cv2.destroyAllWindows()
```

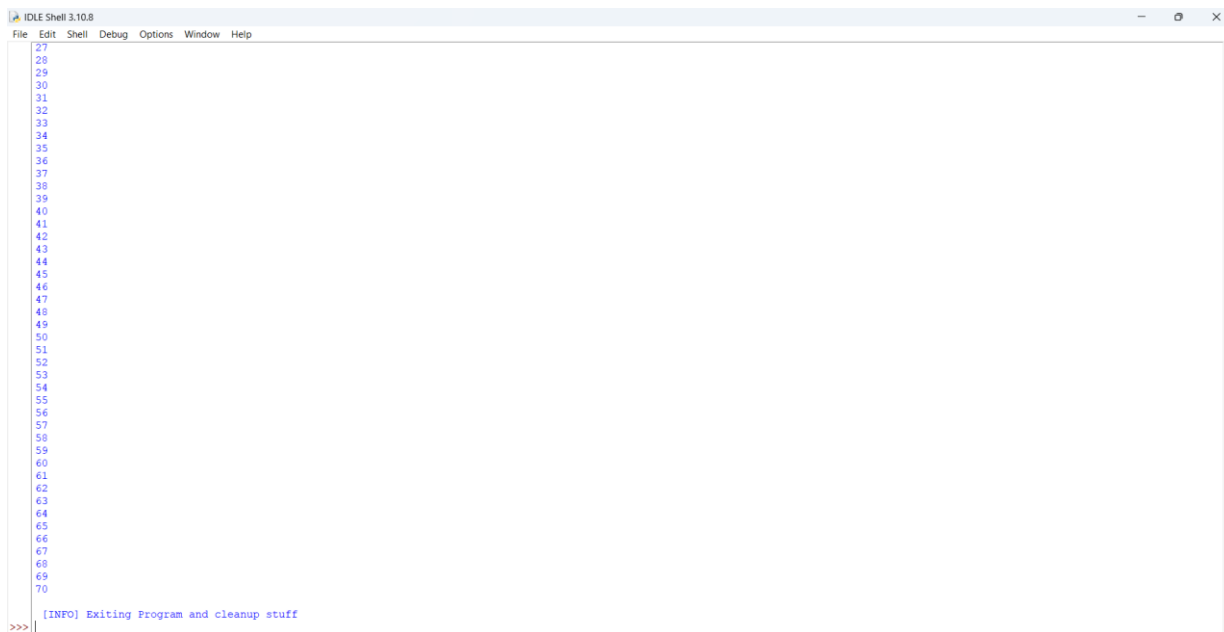
B. SCREEN SHOTS



```
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\kaveyha\OneDrive\Desktop\project(full)\Door bell\Door Bell\face_dataset.py

enter user id end press <return> ==> 1

[INFO] Initializing face capture. Look the camera and wait ...
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
```



```
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70

[INFO] Exiting Program and cleanup stuff
>>>
```

```
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\kaveyha\OneDrive\Desktop\project(full)\Door bell\Door Bell\Door_bell.py
Hello Good to See you Again
```

```
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\kaveyha\OneDrive\Desktop\project(full)\Door bell\Door Bell\Door_bell.py
Hello Good to See you Again
Ready
device online
|
```

```
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\kaveyha\OneDrive\Desktop\project(full)\Door bell\Door Bell\Door_bell.py
Hello Good to See you Again
Ready

device online
checkStat

Start Capturing
sending command
Permission Granted
Opening Door

|
```

```
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\kaveyha\OneDrive\Desktop\project(full)\Door bell\Door Bell\Door_bell.py
Hello Good to See you Again
Ready

device online

|
```

```
"IDLE Shell 3.10.8"
File Edit Shell Debug Options Window Help
Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
- RESTART: C:\Users\kaveyha\OneDrive\Desktop\project(full)\Door bell\Door Bell\Door_bell.py
Hello Good to See you Again
Ready

device online
checkStat

Start Capturing
Permission Denied
Start recording..
Stop recording..
```

C. RESEARCH PAPER

ENHANCED SECURITY USING NEURAL NETWORK FOR PRIVACY AND SECURITY THREAT

Kalari Likitha
UG Scholar
Dept. of IT
Sathyabama Institute of Science
and Technology
Chennai, India
likitha.kalari999@gmail.com

Kaveyha G
UG Scholar
Dept. of IT
Sathyabama Institute of Science
and Technology
Chennai, India
kaveyha842@gmail.com

P Ajitha
Associate Professor
Dept. of IT
Sathyabama Institute of Science
and Technology
Chennai, India
ajithaponnupilai@gmail.com

Abstract— Now-a-days, IOT is reaching its height in the modern era. Home automation is a growing trend as the globe becomes smarter. The most up-to-date home automation technology is the smart door control system. The purpose of this article is to use an Android phone and embedded technology to expand door automation. As soon as the doorbell is touched, the PC starts taking pictures of people and sends the still images to an Android mobile via the Telegram software, alerting the owner to the person at the door. Python and embedded c have been used to establish the full systems programming for a controller. This project focuses on IOT-related communication and security issues. We also show that such archetypes are susceptible to cyber threats. These days, internet of things security systems is gaining popularity.

Keywords: Python, Embedded C, Machine Learning, Face Recognition, Telegram chat-bot, Ardiuno-Uno, Haar- cascade Algorithm, LBPH Algorithm, L298N motor driver, Liquid Crystal Display

I. INTRODUCTION

A new digital revolution is on the horizon, and it will be built on the Internet of Things (IOT) and "Cyber Physical Security Architecture," both of which are included on Gartner's list of the top 10 critical technological trends for 2016. In addition to the services provided by other IoT implementations, smart vehicular networks transform automobiles into powerful sense-and-move platforms that assist with safe navigation, pollution control, reducing the amount of land required for parking, energy

efficiency, shorter delays, passenger comfort, and congestion management through efficient traffic flow. It was in the 20th century when the idea of self-driving cars first gained traction. It wasn't possible

until recent advancements in deep learning, image processing, communication protocols, and route planning, all of which occurred during the previous decade. Autonomous vehicles have many benefits, including increased freedom for the elderly and the disabled, cheaper transportation due to increased car sharing, easier implementation of alternative fuels, less traffic fatalities and injuries, better road usage (fewer roads will be required), less greenhouse gas emissions (more passengers per gallon), and improved disaster response Both the

"Cyber Physical Security Architecture" and the "Internet of Things" both of which are on Gartner's list of the top 10 significant technological trends for 2016 are poised to usher in a new era of digital innovation. Connected to a smart vehicular network, vehicles become powerful sense-and-move platforms that can improve navigational safety, reduce pollution, lessen the need for parking space, use less energy, reduce wait times, improve rider comfort, and better manage traffic congestion. The concept of self-driving automobiles was first proposed in the 20th century. It wasn't possible until recent advancements in deep learning, image processing, communication protocols, and route planning, all of which occurred within the previous decade. Autonomous vehicles have many positive applications, including facilitating greater independence for the elderly and those with physical limitations, reducing the cost of transportation through car sharing, facilitating the introduction of alternative fuels, enhancing road use (fewer roads will be needed), climate change mitigation (more people per mile per gallon), and disaster response. Distractions from cellphones are at blame for 18% of collisions that result in injuries, a problem that is particularly common among younger generations.

The communication and security issues posed by IOT are the main topics of this research. Additionally, we show how such models might be vulnerable to cyber-attacks. The internet of things for security systems is now gaining popularity. Using embedded technology and an Android phone, this article aims to expand the door automation approach. As soon as the doorbell is pressed, the computer starts to take a picture of the person at the door and transmits the still photo to the owner's Android mobile via the Telegram app. Python and embedded C for a controller have been used to establish the full systems programming

II. LITERATURE SURVEY

A literature review is a body of writing that seeks to summarize the key ideas in the field's current understanding and/or methodological approaches to a certain subject. It's secondary sources that talk about stuff that's already been published in a given field, and sometimes it talks about stuff that's already been published in a certain field during a certain time frame. Its primary purpose is to provide the reader with a summary of the most current findings in a particular area of study, and it also lays

the groundwork for other goals, such as subsequent research. An annotated bibliography might serve as this step's equivalent. It has a format and often incorporates synthesis and summary. Difference between a summary and a synthesis is that the latter just restates the source's most important points, while the former reorganizes and restructures the information. It may provide a new perspective on established findings, combine new and old ideas, or trace the evolution of the field's thinking across time, highlighting important debates along the way. Depending on the context, the literature review may provide an evaluation of the available sources and recommendations on which are the most useful to the reader.

The Automatic Face Recognition Door Access System was developed by Hteik Htar Lwin, Aung Soe Khaing, and Hla Myo Tun. The MATLAB computer program is utilized in this technique to recognize people instantly upon their appearance. Using incoming data received from the PC, the microcontroller regulates the door access mechanism. After authenticating the identity of the visitor, the entryway is opened rapidly. The entryway consequently closes following two seconds. For a person to enter, however, two seconds is insufficient. Therefore, a longer duration should be chosen to reflect the current situation. Face detection uses the Viola Jones algorithm. The frontal aspect of the face is the only part of the face that this algorithm can accurately recognize, hence this technology has limits.

- Using Internet of Things technology, researchers Sandesh Kulkarni, Minakshee Bagul, Akansha Dukare, and Prof. Archana Gaikwad proposed a "Face Recognition System." Cost- and energy-efficient, the system was designed. The Viola Jones Algorithm is employed in this system for face identification, which removes face candidates via a series of steps, however its ability to collect structural data is rather constrained

- An effective face detection method was put out by Paul Viola. The notion of integral images and the Adaboost classifier, which produces quick and accurate face identification, are introduced in this study.

- We realized that, in terms of discriminative power, LBPH (Local Binary Pattern Histogram) outperforms Viola Jones. So, in addition to the frontal view, a person may be recognized by their diverse poses and lighting.

- Security of residential doors is crucial, serves as the foundation for the simplest and easiest security, and is sufficient to give homeowners a sense of security. As technology has advanced, particularly in the IOT field, so have methods for locking residential doors, such as locking residential doors with faces and others. Facial recognition technology has also advanced and been used to house door locking systems; it is a simple, user-

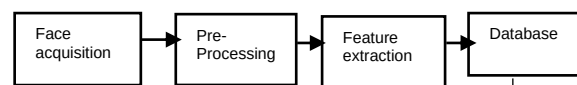
friendly solution that is fairly accurate in detecting the faces of homeowners. The CNN technique for facial identification was developed, and it has since emerged as one of the face recognition systems that is simple to use, has excellent accuracy in detecting faces, and has been used in object recognition systems and others. The CNN Alex net facial recognition system, which is integrated into a door locking system, was used in this study to collect data. The system used to collect the 1048 facial data on the homeowner's face was then used to train machine learning, which produced results that were quite accurate (97.5% accuracy), which is good compared to some other studies. The CNN Alex net approach, which can be used to the IOT device known as the Raspberry Pi, can do facial recognition pretty accurately.

III. EXISTING SYSTEM

Our goal is to give the customer a live footage-based approval system connected by IOT and a Telegram chat-bot. The open problems in the present system are that practically all of the solutions are built on Raspberry Pi and other Microcontrollers, which perform better on the upper side.

IV. PROPOSED SYSTEM

Every day, people's interest in security and safety grows. It is thus being enhanced and used to make our lives easier. The security of one's home, workplace, or organization must keep up with how technology has merged into people's daily life. A system with little human involvement is what we're aiming for. Developing a smart receptionist with a smart lock system was primarily done for security and safety reasons. When the business or establishment's main entrance is shut, this clever security mechanism is employed to detect a visitor. This system's function is to use the Telegram application to manage authorization and the door lock. In this setup, the camera records the person's image each time they ring the bell. If the person's image is already in the database, entry will be granted; otherwise, the smart lock system will transmit the person's image to the Telegram App for authorization. The option for the user to lock or unlock the door will be accessible once the image has been transmitted to the Telegram application. A person standing at the door may enter if the user presses the unlock button, whereas if the user clicks the lock button, the door will remain locked and a person may not enter.



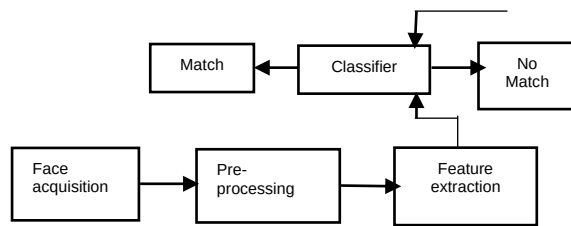


Figure 1 Architecture of Proposed Model

In order to improve the efficacy, efficiency, and delivery of real-time solutions, we integrated this powerful machine learning tool with the hardware (Embedded Systems) in our project program, which was written in Python. Facial recognition plays a significant part in our project. We have a button linked to the project's microcontroller, and when the button is pressed, the camera activates and begins recognizing faces using the Haarcascade and LBPH algorithms.

- If a match is found, Python sends the relevant information to an Arduino Uno microcontroller, which then triggers an automated door unlock. To record film and send it to the Telegram mobile app through Bot Father if a face is not identified, the system checks for the presence of a camera. To get the result to the microcontroller, a computer uses serial connection.

- Once that procedure is finished, the user has the choice to authenticate or not. In the event of a successful authentication, the door will open; otherwise, it will remain closed.

V. WORKING

We are dividing this project into two sectors one is the software part and another is the Hardware Part.

Explanation

Software

Trained Model will be used to identify individuals by comparing them to the training dataset, which was generated using the face-recognition method LBPH (Local Binary Pattern Histogram).

Hardware

A trigger is wired into the project's microprocessor for use in this context. The camera is triggered upon button push, and subsequent face detection is carried out with the help of the Haar cascade and the LBPH

algorithm; the results are then sent to the microcontroller via serial communication, which is linked to the computer.

a) Process Methodology

1. Haar Cascade

A wavelet family or basis is a group of related scaled-down "square-shaped" functions. The picture's pixels are broken up into rectangles based on their roles using the Haar Wavelet transform. A high degree of accuracy is achieved through the application of machine learning methods to "training data." The "features" that are identified are determined by applying "integral image" concepts to the data. To achieve accurate classification output, Haar Cascades employs the Ada boost learning method, which chooses a few crucial features from a large pool.

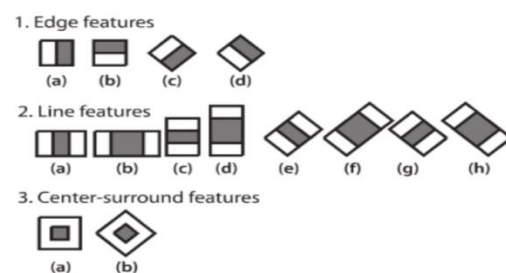


Figure 2. Haar Cascade

Face Detection: pinpoints where and how big human faces are in random (digital) photos.

Face Detection: It is used before Face Recognition to identify and isolate a face before it can be identified.

2. Feature Extraction

Haar Fountains use AI methods to prepare a capability on a major arrangement of positive and negative instances of a given item. The feature extraction stage of the algorithm looks like this.

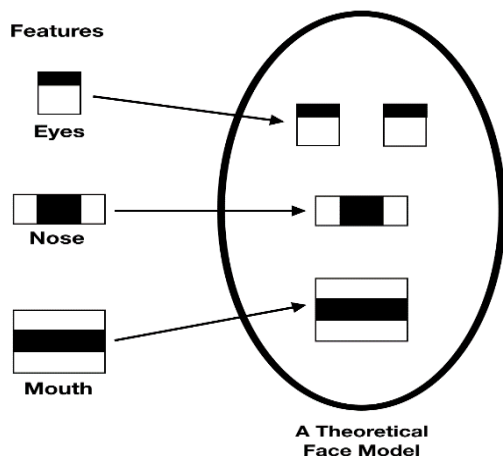


Figure 3 Feature Extraction

3. L298 N Motor Driver

The L298N is a dual H-Bridge motor controller that enables simultaneous adjustment of two DC motors' rotational speeds and directions. The maximum peak current the module can supply is 2A, and the voltage range it can power DC motors with is 5V to 35V. The answer to this question is found in the VCC voltage of the motor. The gadget includes a 5V converter that can be switched on or off via a resistor. To use the 5V regulator's 5V port as an output, for example to charge our Arduino board, the motor source voltage must be at least 12V.

4. Arduino UNO

The ATmega328P forms the basis of the Arduino UNO microprocessor device. It has a USB port, a power connector, an ICSP interface, a restart switch, and 14 digital I/O ports (6 of which can be used as PWM outputs). It also has a 16 MHz ceramic resonator. It has everything you need to get started with the microprocessor; all you need is a USB connection to link it to a computer or an AC-to-DC converter or battery to charge it. You can experiment with your UNO without too much fear of breaking it;

in the worst-case situation, you'll only be out a few bucks and a new chip.

5. Liquid Crystal Display

Alphanumeric liquid crystal screens that enable for conversation (LCDs). Most text-based LCDs use a microprocessor made by Hitachi called HD44780, and this software makes it possible for an Arduino or Genuino device to interact with such screens.

6. 9V Battery

The nine-volt battery, also known simply as the 9-volt battery, is a type of rechargeable electronic battery with a standard energy output of 9 volts. The range of "real" voltage between 7.2 and 9.6 volts varies by battery composition. There are many different types of batteries on the market, but one of the most popular is the PP3 battery, which was originally designed for use in early pocket devices.

7. USB Cable

You can use it to hook up your Arduino Uno, Mega 2560, 101, or any other board to the computer's USB female A connection. About a metre of cable is included. The actual cables you receive may have a slightly different form or colour than the one pictured. Type A/B USB 2.0 cables are the industry standard.

8. Push Button

Pushbuttons and switches are used to make a connection between two locations in a circuit. With this code, pressing the trigger activates the onboard LED connected to port 13.

9. DVD Loader

This is used as a sample door for this project. The automatic opening and closing of the dvd loader symbolizes the automatic moment of the door. As we give the instructions the dvd loader acts according to the instructions given.

BLOCK DIAGRAM

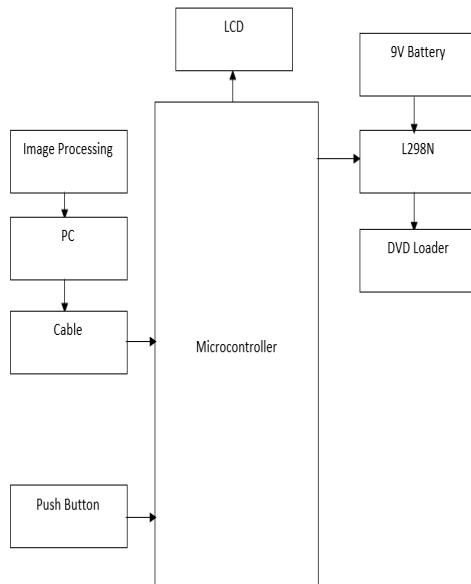


Figure 4 Block Diagram

OUTPUT

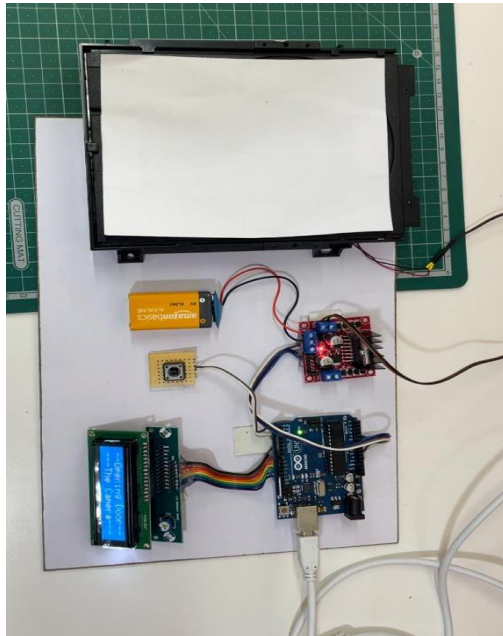


Figure 5 Hardware Module

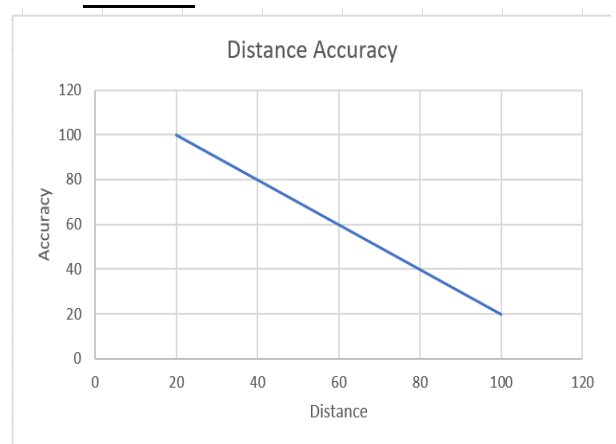
VI. FUTURE WORK

This smart door locking and unlocking system is of wire based. This can be further implemented into wireless by using raspberry-pi. The codes and the connections are stored in raspberry-pi.

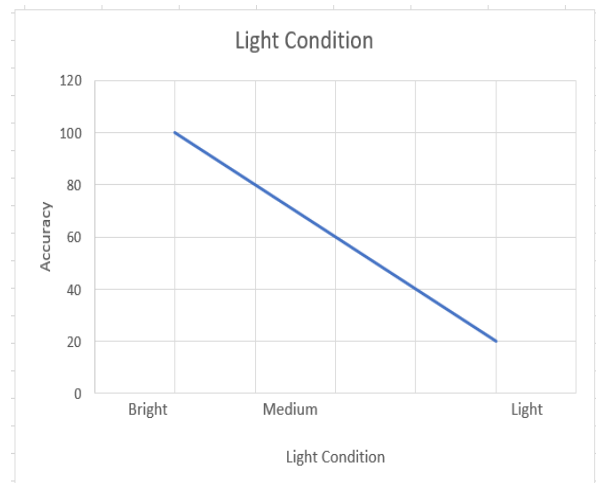
VII. Result

The primary goal of the project is to develop a reliable IOT-based security solution that can be used in a variety of scenarios, such as in bank locker rooms, women's hostels, and confidential meeting rooms, to authenticate only authorized people using embedded 'c' and Python language integrated with Telegram Chat-Bot.

Distance-Accuracy:



Distance-Light Condition:



VIII. REFERENCES

- [1] Image Classification Using Convolutional Neural Networks: A step by step guide, Devansh Sharma — January 11, 2021

- [2] NV SUMANTH KAUSHIK, SHREYAS D G, GURURAJA H S, SMART VIDEO SURVEILLANCE SYSTEM, ISSN:1001-1749, Volume XVI, Issue VI, JUNE/2020

- [3] Ishan Kokadwar, Anurag Kulkarni, SayaliKhare, Vaibhav Limbhore, Swati Chandurkar, Camera based Smart Surveillance System-Literature Survey, Volume: 07 Issue: 06 | June 2020

- [4] Prasanna Rajendra, NiketWorlikar, Yash Mahajan, Siddhesh Swami, Smart Surveillance Using OpenCV, Motion Analysis and Facial Landmark, - Volume 7 Issue 1 - Jan to April 2020.

- [5] Dayana R, Suganya M, Balaji P, Mohamed Thahir A, ArunkumarP, Smart Surveillance System using Deep Learning, ISSN: 2277-3878, Volume-9, Issue-1, May2.

