IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 1, January 2025

DOI: 10.17148/IJARCCE.2025.14125

IMPLEMENTATION OF OPEN SOURCE IOT BASED SMART POLE HOME AUTOMATION SYSTEM

Ms Anitha Kumari S[1], Lingraj H[2], Mahesh babu[3], Manjunatha G[4], Nishanth D[5]

Associate Professor, ECE, East West Institute of Technology, Bengaluru, India[1]

Student, ECE, East West Institute of Technology, Bengaluru, India[2]

Student, ECE, East West Institute of Technology, Bengaluru, India[3]

Student, ECE, East West Institute of Technology, Bengaluru, India[4]

Student, ECE, East West Institute of Technology, Bengaluru, India[5]

Abstract: This project presents the design and implementation of a Smart Pole Home Security System utilizing Arduino and various sensors to provide an automated, efficient, and secure solution for home safety and control. The system integrates key components including an LDR for day/night detection, fire and gas sensors for hazard detection, a DC motor for automated gate control, and relays for home appliance management (e.g., lights and fans). A laptop camera is employed for vehicle number plate recognition and face recognition, which serve as a means of authenticating visitors and vehicles approaching the home. The system leverages the Telegram Bot API to communicate with the homeowner, sending real-time alerts and photos when unauthorized access is detected. The homeowner can then remotely grant or deny access via the Telegram app. If access is denied, or if the system identifies a threat, it automatically sends an alert to the nearest police station. The project enhances home automation by controlling lights and fans based on day/night conditions and provides a higher level of security through real-time monitoring and remote intervention.

Keywords: gas detection, water detection, Internet of Things (IoT), machine learning.

I. INTRODUCTION

In an era of increasing technological advancement, the need for smarter, more secure home systems has become crucial. Traditional security methods, such as manual locks and basic alarm systems, are no longer sufficient to ensure comprehensive home safety. With the rise of automation and the Internet of Things (IoT), integrating smart systems into daily life has become a viable solution for homeowners seeking enhanced convenience, control, and security. This project introduces a Smart Pole Home Security System, which leverages modern sensors, actuators, and communication technologies to create an automated security and home control system. The system is based on Arduino and integrates components such as LDRs, fire and gas sensors, a DC motor, and relays for controlling home appliances like bulbs and fans. In addition to hardware automation, it employs a laptop camera for vehicle number plate recognition and face recognition, enabling the system to authenticate people and vehicles entering the premises.

A key feature of this system is its ability to communicate with the homeowner via the Telegram Bot API. When an unauthorized vehicle or person is detected, the system sends real-time alerts, including images, to the homeowner's smartphone through the Telegram app. This allows the homeowner to remotely monitor and control access to their property. If the homeowner denies access or does not respond, the system automatically alerts the local authorities, further enhancing security. The use of LDRs for day and night detection helps automate lighting, while fire and gas sensors continuously monitor the environment for safety hazards.

The DC motor allows for automated gate control based on authentication, providing both convenience and security. The project's primary goal is to develop a cost-effective, reliable, and scalable smart home security system that integrates automation, real-time monitoring, and remote control. By combining simple hardware components with advanced software tools, this system offers a solution for modern security challenges in residential environments. The integration of IoT, image recognition, and cloud-based communication through Telegram makes it adaptable to various applications and enhances overall home safety and convenience

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 1, January 2025
DOI: 10.17148/IJARCCE.2025.14125

II. LITERATURE REVIEW

This research designs face detection and recognition systems for smart home security application. The design is implemented using MyRIO 1900 and programmed using LabVIEW. The connection between myRIO and computer is wifi network. The image of a person is acquired via webcam connected to MyRIO using USB cable. The face detection system is built based on the template matching, while the face recognition is based on the principle component analysis. The testing is done to examine the performance of the face detection in various change of distance, light intensity, light position angles, person's accessories and shirt colour. The face detection modul has good performance in some conditions as distance between the person and the camera is less than 240 cm, person doesn't use accessories that cover part of face, person doesn't use shirt with colour similar to skin colour, and background colour is difference from skin colour. While the face recognition system has 80% of accuracy when it is tested using realtime image. The combination with password is needed in order to increase the security level as it is applied in real smart home security systems.

Published in: 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)
Date of Conference: 01-02 November 2017
Date Added to IEEE *Xplore*: 08 February 2018
ISBN Information:
DOI: 10.1109/ICITISEE.2017.8285524
Publisher: IEEE
Conference Location: Yogyakarta, Indonesia

Author:
Dwi Ana Ratna Wati
Universitas Islam Indonesia, Yogyakarta, Daerah Istimewa Yogyakart, ID
Dika Abadianto
Department of Electrical Engineering, Universitas Islam Indonesia, Yogyakarta, Indonesia

Development of Real-Time Face Recognition for Smart Door Lock Security System using Haar Cascade and OpenCV LBPH Face Recognizer
Face recognition is a technology that is widely used in security systems. In a door security system, facial recognition can be used to open the door simply by recognizing the face of the door owner. This study aims to develop a real-time facial recognition system for smart locking doors using Haar Cascade and OpenCV LBPH Face Recognizer. The purpose of this project is creating security system to limit people who can access a room. The Haar Cascade method is used to detect faces in images, while the OpenCV LBPH Face Recognizer is used to recognize detected faces. This system was developed using the Python programming language and the OpenCV library. The test results show that this system can detect and recognize faces with an accuracy of 62.7% with our dataset and can be improved by adding more datasets and using deep learning algorithms to train the recognizer model. Thus, the developed real-time facial recognition system can be used as a smart locking door security solution with high accuracy.

Published in: 2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)
Date of Conference: 16-16 February 2023
Date Added to IEEE *Xplore*: 23 May 2023
ISBN Information:
DOI: 10.1109/ICCoSITE57641.2023.10127753
Publisher: IEEE
Conference Location: Jakarta, Indonesia

Authors
Daniel Anando Wangean
Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia
Sinjiru Setyawan
Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia
Fairuz Iqbal Maulana
Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia
Gusti Pangestu
Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia
Choirul Huda
Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia

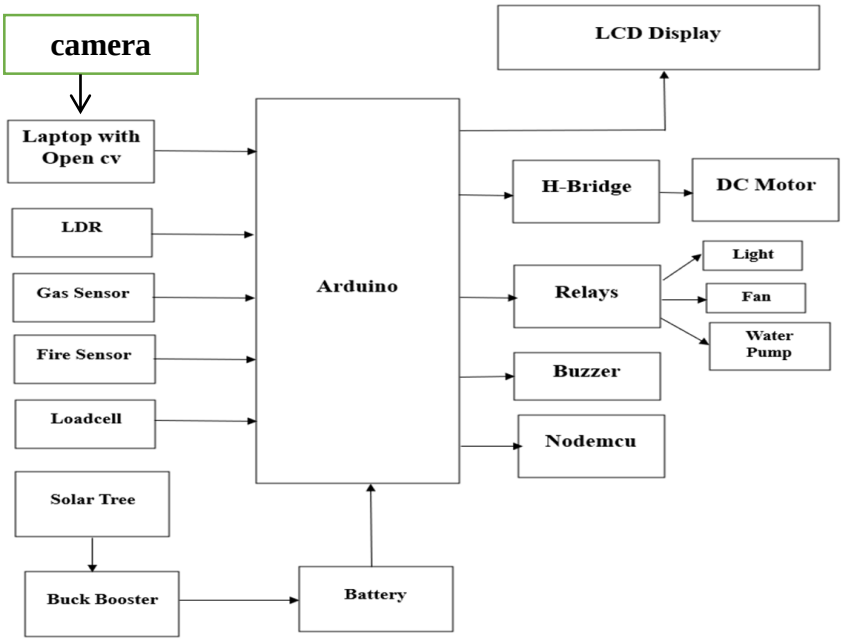
IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 1, January 2025

DOI: 10.17148/IJARCCE.2025.14125

III. METHODOLOGY



IV. WORKING

1. Day/Night Detection and Lighting Control (LDR Sensor)

The LDR sensor continuously monitors ambient light conditions to detect whether it is day or night.
During nighttime, the LDR automatically turns on outdoor lights through a relay. Conversely, during the day, the lights are turned off to conserve energy.
This feature automates the lighting system based on real-time conditions without requiring manual intervention.

2. Automated Gate Control with Number Plate and Face Recognition

When a vehicle approaches the home gate, the laptop camera captures an image of the vehicle's number plate.
The captured image is processed using image recognition algorithms to identify whether the vehicle's number plate matches any pre-authorized numbers stored in the system's database.
Simultaneously, the camera captures the visitor's face, and the system uses face recognition algorithms to identify whether the person is authorized.
If both the vehicle and the person are recognized as authorized, the DC motor automatically opens the gate, allowing entry.
If the system fails to recognize the vehicle or the person, it sends an alert to the homeowner via Telegram.

3.Visitor Authentication and Remote Access Control

In case of unauthorized detection, the system captures an image of the visitor and sends it to the homeowner through the Telegram Bot.
The homeowner can view the captured image and decide whether to grant or deny access remotely.
If the homeowner grants permission via Telegram, the gate will open automatically.
If the homeowner denies access, the system sends an alert to the local police station, along with the captured images of the vehicle and the visitor, notifying authorities of the potential security threat.

3. Fire and Gas Hazard Detection

The system integrates fire and gas sensors to continuously monitor the environment for hazardous conditions, such as a fire outbreak or gas leak.
If the fire sensor detects smoke or a rise in temperature, or if the gas sensor detects gas leakage, an immediate alert is sent to the homeowner via Telegram.
The system can also be configured to activate safety protocols such as triggering an alarm or turning off the gas supply (if applicable) using relays.
This provides the homeowner with real-time hazard detection and allows for quick response to emergencies.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 1, January 2025
DOI: 10.17148/IJARCCE.2025.14125

4. Home Automation for Appliance Control

The system is equipped with relays to control home appliances such as bulbs and fans.
The appliances can be automatically turned on or off based on certain conditions (such as day/night detection through the LDR) or manually controlled by the homeowner through the Telegram app.
This enhances convenience by integrating home automation with the security system, enabling the homeowner to manage both security and comfort remotely.

5. Real-time Alerts and Communication via Telegram

The Telegram Bot API is used to send real-time alerts and notifications to the homeowner's smartphone.
The system captures images or videos of any unauthorized activity and sends them as alerts, allowing the homeowner to monitor the situation remotely.
The homeowner can respond to these alerts and control the system (e.g., grant access, turn on/off appliances) directly from the Telegram app.
In case of emergencies, the system can also notify local authorities if unauthorized access is detected and denied by the homeowner.

6. Emergency Response and Law Enforcement Notification

If the system detects unauthorized access and the homeowner denies entry through the Telegram app, the system automatically sends the captured images and details to the nearest police station.
This feature provides an additional layer of security, ensuring timely intervention in case of a security breach or threat.

V. HARDWARE AND SOFTWARE DESCRIPTIONS

Hardware Requirements:

- **Arduino**
- **Raspberry pi 4**
- **LCD Display**
- **LDR Sensor**
- **Fire Sensor**
- **Gas Sensor**
- **H-Bridge**
- **DC Motor**
- **Power Supply**

Software Requirements:

- **Arduino IDE**
- **Embedded C**

Hardware Requirements:

Arduino mega 2560:

The **Arduino Mega 2560** is a versatile and powerful microcontroller board based on the **ATmega2560** chip. It is widely used in projects that require a large number of input/output (I/O) pins, more memory, and advanced communication capabilities.

Key specifications include:

- **Microcontroller:** ATmega2560
- **Digital I/O Pins:** 54 (15 PWM capable)
- **Analog Inputs:** 16
- **PWM Pins:** 15
- **Flash Memory:** 256 KB
- **SRAM:** 8 KB
- **EEPROM:** 4 KB
- **Clock Speed:** 16 MHz
- **USB Interface:** Used for programming and serial communication
- **Communication:** Supports UART, I2C, and SPI interfaces

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 1, January 2025

DOI: 10.17148/IJARCCE.2025.14125

The Mega 2560 operates at 5V and can be powered through a USB connection, an external power jack (7-12V), or the Vin pin. It offers a large number of I/O pins, making it ideal for projects like robotics, 3D printers, and home automation systems that require many sensors or actuators.

The board is compatible with the **Arduino IDE**, making it easy to program in C/C++. It also comes with a pre-installed bootloader, simplifying the process of uploading code. Thanks to its ample resources, the Arduino Mega 2560 is widely used in advanced, complex applications requiring a lot of interfacing and control.

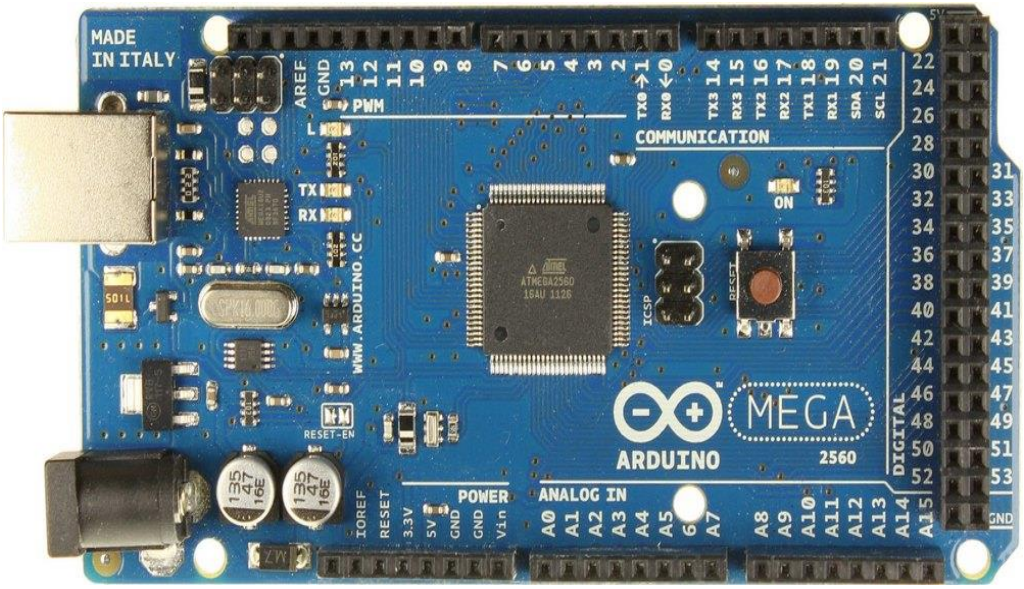


Fig:1. Arduino mega 2560

Raspberry pi 4:

The **Raspberry Pi 4 Model B** is a powerful single-board computer designed for a wide range of applications, from education to DIY projects, and even industrial uses. It is the latest and most powerful iteration of the Raspberry Pi family, offering significant improvements over previous models.

Key features of the Raspberry Pi 4 include:

- **Processor:** Broadcom BCM2711, Quad-Core ARM Cortex-A72 (64-bit) running at 1.5 GHz
- **RAM Options:** 2GB, 4GB, or 8GB LPDDR4-3200 SDRAM
- **Graphics:** VideoCore VI GPU, supporting 4K output (2 monitors via HDMI)
- **Storage:** microSD card slot for OS and data storage
- **Networking:** Gigabit Ethernet, dual-band 802.11ac Wi-Fi, Bluetooth 5.0
- **USB Ports:** 2 USB 3.0 ports, 2 USB 2.0 ports
- **GPIO Pins:** 40-pin GPIO header, fully backward-compatible with previous Raspberry Pi models
- **Power Supply:** 5V/3A USB-C power input

The Pi 4 supports high-definition video, faster networking, and improved processing power, making it suitable for projects like media centers, home automation, robotics, and web servers. It runs a variety of operating systems, with **Raspberry Pi OS** being the most common.

With its powerful performance and versatility, the Raspberry Pi 4 is an excellent choice for both beginner and advanced users in the maker and developer communities.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 1, January 2025

DOI: 10.17148/IJARCCE.2025.14125

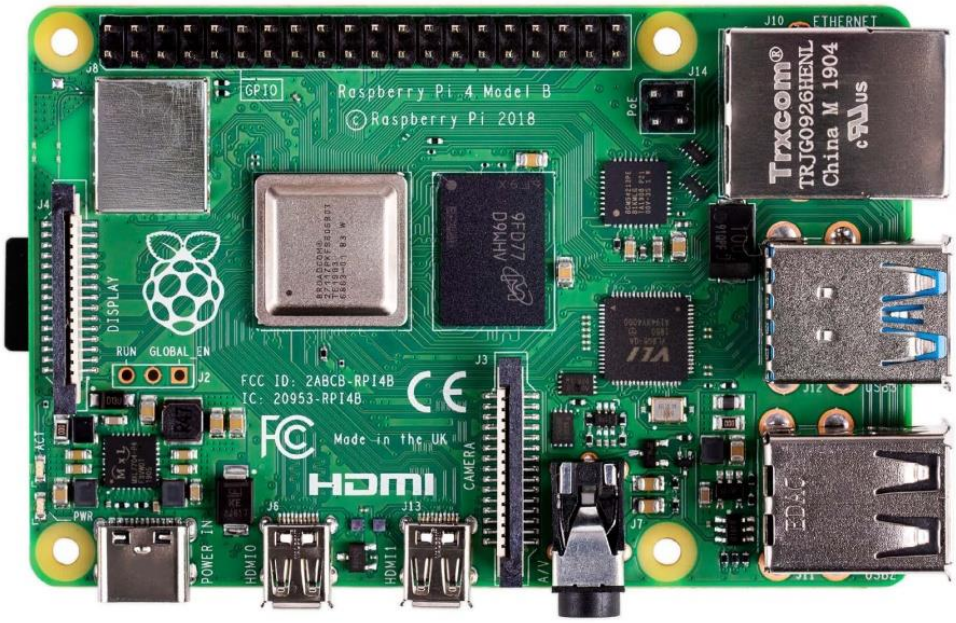


Fig:2 Raspberry pi 4

Software Requirements:

Arduino mega

A program for Arduino may be written in any programming language for a compiler that produces binary machine code for the target processor. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio.

The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus.

A program written with the IDE for Arduino is called a sketch. Sketches are saved on the development computer as text files with the file extension .ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension .pde.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main() into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution.

A minimal Arduino C/C++ sketch, as seen by the Arduino IDE programmer, consist of only two functions:

- **setup():** This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.

- **loop():** After setup() has been called, function loop() is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 1, January 2025

DOI: 10.17148/IJARCCE.2025.14125

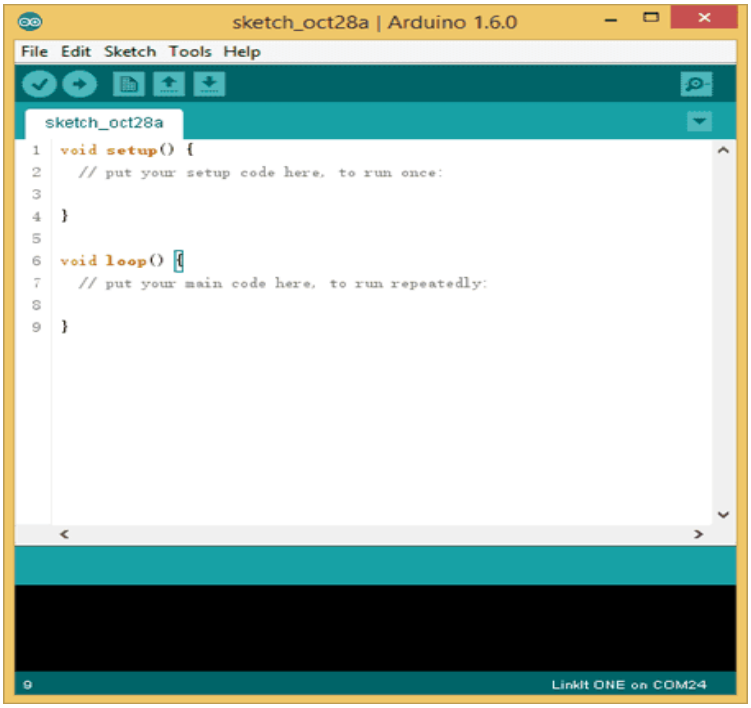


Fig . An Arduino Sketch

Embedded C:

- When designing software for a smaller embedded system with the 8051, it is very common place to develop the entire product using assembly code. With many projects, this is a feasible approach since the amount of code that must be generated is typically less than 8 kilobytes and is relatively simple in nature. If a hardware engineer is tasked with designing both the hardware and the software, he or she will frequently be tempted to write the software in assembly language.

- The trouble with projects done with assembly code can is that they can be difficult to read and maintain, especially if they are not well commented. Additionally, the amount of code reusable from a typical assembly language project is usually very low. Use of a higher-level language like C can directly address these issues. A program written in C is easier to read than an assembly program.

- Since a C program possesses greater structure, it is easier to understand and maintain. Because of its modularity, a C program can better lend itself to reuse of code from project to project. The division of code into functions will force better structure of the software and lead to functions that can be taken from one project and used in another, thus reducing overall development time. A high order language such as C allows a developer to write code, which resembles a human's thought process more closely than does the equivalent assembly code. [25]The developer can focus more time on designing the algorithms of the system rather than having to concentrate on their individual implementation. This will greatly reduce development time and lower debugging time since the code is more understandable.

- By using a language like C, the programmer does not have to be intimately familiar with the architecture of the processor. This means that someone new to a given processor can get a project up and running quicker, since the internals and organization of the target processor do not have to be learned. Additionally, code developed in C will be more portable to other systems than code developed in assembly. Many target processors have C compilers available, which support ANSI C.

- All of this is not to say that assembly language does not have its place. In fact, many embedded systems (particularly real time systems) have a combination of C and assembly code. For time critical operations, assembly code is frequently the only way to go. One of the great things about the C language is that it allows you to perform low-level manipulations of the hardware if need be, yet provides you the functionality and abstraction of a higher order language.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 1, January 2025

DOI: 10.17148/IJARCCE.2025.14125

PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is interpreted:** Python is processed at runtime by the interpreter.You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive:** You can actually sit at a Python prompt and interactwith the interpreter directly to write your programs.
- **Python is Object-Oriented:** Python supports Object-Oriented style ortechnique of programming that encapsulates code within objects.
- **Python is a Beginner's Language:** Python is a great language for thebeginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Python Features

- **Easy-to-learn:** Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read:** Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain:** Python's source code is fairly easy-to-maintain.
- **A broad standard library:** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode:** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable:** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable:** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases:** Python provides interfaces to all major commercial databases.
- **GUI Programming:** Python supports GUI applications that can be created and ported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable:** Python provides a better structure and support for large programs than shell scripting.

Python's standard library

- Pandas
- Numpy
- Sklearn
- Seaborn
- Matplotlib
- Importing Datasets
- OpenCV

IMAGE PROCESSING USING OPENCV IN PYTHON

1) Image processing is the process of manipulating pixel data in order to make it suitable for computer vision applications or to make it suitable to present it to humans. For example, changing brightness or contrast is an image processing task which make the image visually pleasing for humans or suitable for further processing for a certain computer vision application.

PYTHON

It is an object-oriented programming language. The processing happens during the runtime, and this is performed by the interpreter. Python's simple to learn and easy to use is an advantage and thus makes it developer friendly. It is easier to read and understand as the syntax is conventional. The code can be executed line by line using the interpreter. Python can support multiple platforms like Linux, UNIX, windows, Macintosh, and so on.

The paradigms of Object-oriented programming are supported by python. The functions such as polymorphism, operator overloading and multiple inheritance is supported python.

Open CV

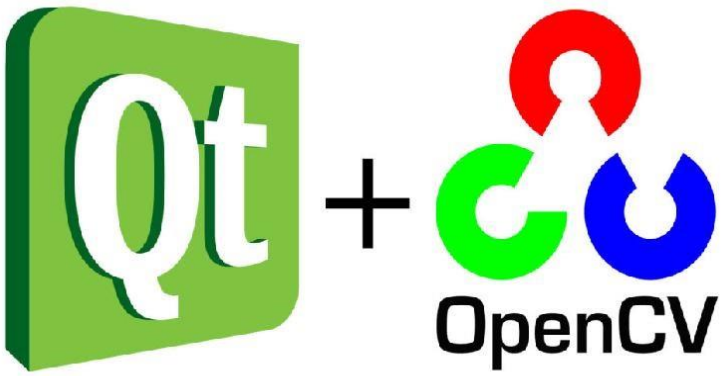
OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is crossplatform and free for use under the open-source BSD license. OpenCV supports deep learning frameworks TensorFlow, Torch/PyTorch and Cafe.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 1, January 2025
DOI: 10.17148/IJARCCE.2025.14125

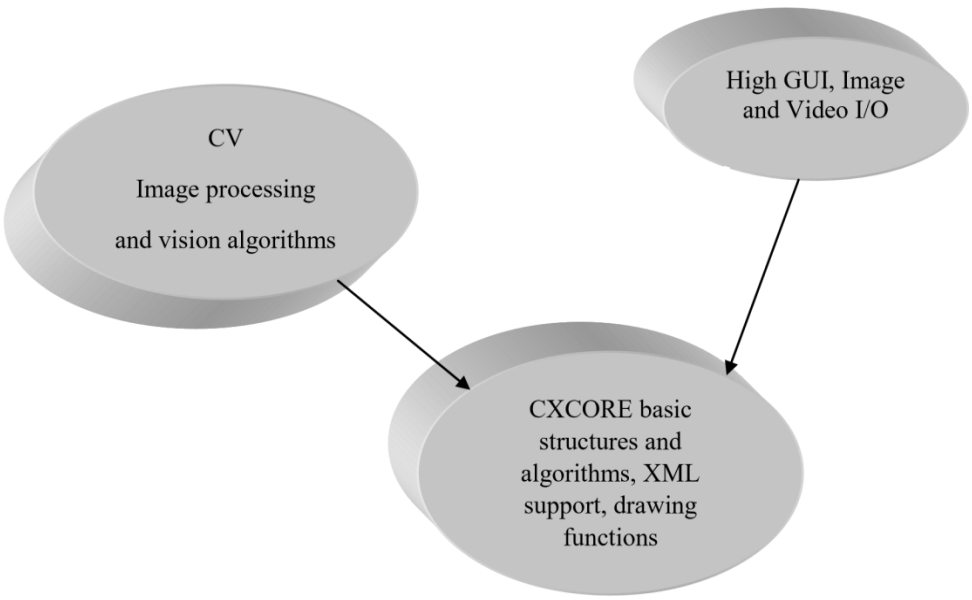
It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

In 1999, the OpenCV project was initially an Intel Research initiative to advance CPUintensive applications, part of a series of projects including real-time ray tracing and 3D display walls. The main contributors to the project included a number of optimization experts in Intel Russia, as well as Intel‘s Performance Library Team. In the early days of OpenCV, the goals of the project were described as:

- Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel.
- Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.
- Advance vision-based commercial applications by making portable, performance optimized code available for free – with a license that did not require code to be open or free itself.



Structure of open cv



Once OpenCV is installed, the OPENCV_BUILD\install directory will be populated with three types of files:

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 1, January 2025
DOI: 10.17148/IJARCCE.2025.14125

- **Header files**: These are located in the OPENCV_BUILD\install\includesubdirectory and are used to develop new projects with OpenCV.
- **Library binaries**: These are static or dynamic libraries (depending on the option selected with CMake) with the functionality of each of the OpenCV modules. They are located in the bin subdirectory (for example, x64\mingw\bin when the GNU compiler is used).
- **Sample binaries**: These are executables with examples that use the libraries. The sources for these samples can be found in the source package.

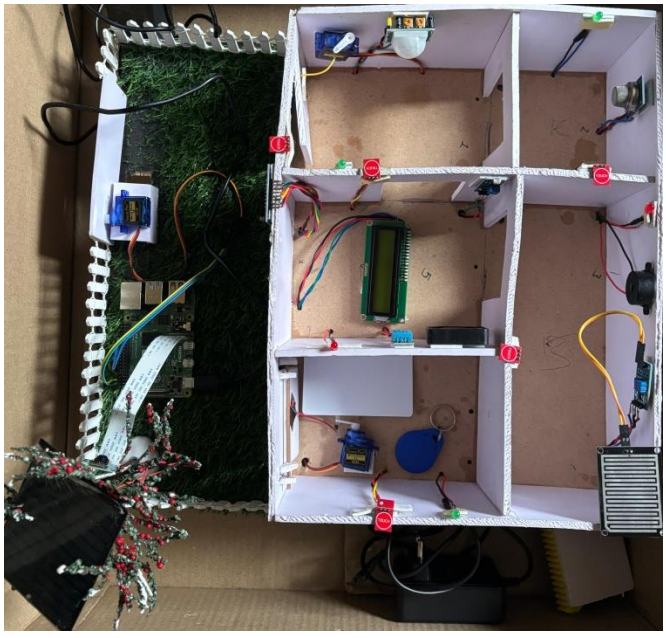
General description

- Open source computer vision library in C/C++.
- Optimized and intended for real-time applications.
- OS/hardware/window-manager independent.
- Generic image/video loading, saving, and acquisition.
- Both low and high level API.

Features

- Image data manipulation (allocation, release, copying, setting, conversion).
- Image and video I/O (file and camera based input, image/video file output).
- Matrix and vector manipulation and linear algebra routines (products, solvers,, SVD).
- Various dynamic data structures (lists, queues, sets, trees, graphs).
- Basic image processing (filtering, edge detection, corner detection, sampling and interpolation, color conversion, morphological operations, histograms, image pyramids).
- Structural analysis (connected components, contour processing, distance transform, various moments, template matching, Hough transform, polygonal approximation, line fitting, ellipse fitting, Delaunay triangulation).
- Camera calibration (finding and tracking calibration patterns, calibration, fundamental matrix estimation, homography estimation, stereo correspondence).
- Motion analysis (optical flow, motion segmentation, tracking).
- Object recognition (eigen-methods, HMM).
- Basic GUI (display image/video, keyboard and mouse handling, scroll-bars).
- Image labeling (line, conic, polygon, text drawing)

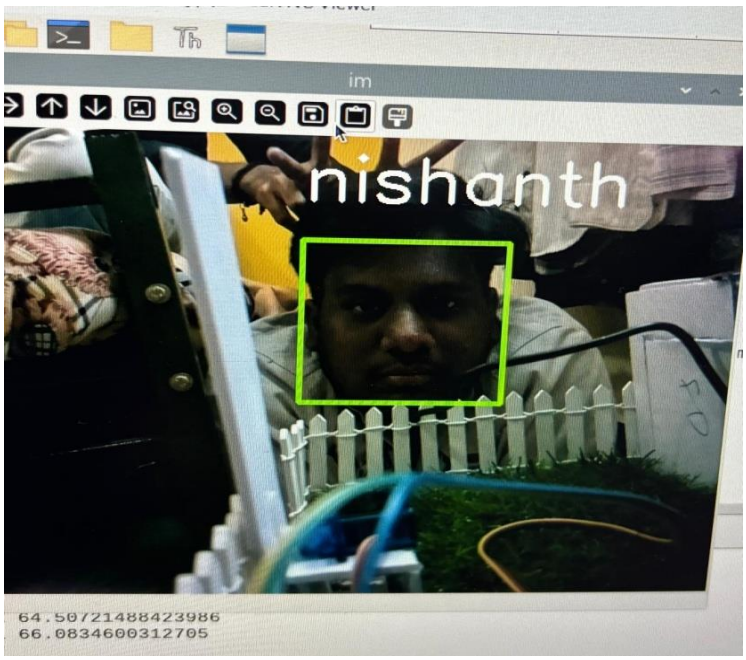
VI. RESULT



Complete model

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 1, January 2025
DOI: 10.17148/IJARCCE.2025.14125

Readings from working model



Known Person detected and Gate Opened Automatically

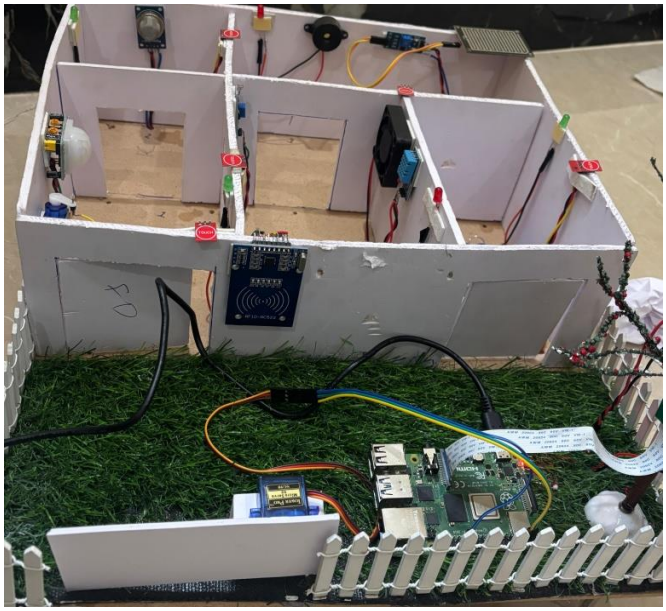
IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 1, January 2025

DOI: 10.17148/IJARCCE.2025.14125

Received alert message via Telegram



When Unknown person detects the gates will be closed

VII. CONCLUSION

This project presents an innovative approach to stress detection and management by offering a proactive solution for monitoring mental health and well-being. The IoT-based stress detection system developed for mental health and well-being integrates advanced technologies to offer a comprehensive and innovative solution for stress monitoring. By incorporating sweat, temperature, and heartbeat sensors, the system provides real-time physiological assessments, detecting stress when threshold values are exceeded. The integration of facial emotion recognition further personalizes the system, analyzing emotional states to validate stress detection. Additionally, the system sends timely notifications and precautionary measures through Telegram, ensuring immediate communication and support. This multifaceted approach combines biometric data, emotion analysis, and effective communication to proactively address stress and promote mental well-being.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 1, January 2025

DOI: 10.17148/IJARCCE.2025.14125

REFERENCES

[1]. Using Face Recognition
[2]. and IoT. International Journal of Innovative Research in Science, Engineering and Technology, 9(7), 8894-8900. Dhage, S., Kakde, A., Kadam, A., & Khaparde, N. (2017). IoT-enabled Smart Home Security System. International Journal of Computer Science and Information Technologies, 8(2), 236-239.
[3]. Srikanth, M., Ramesh, S., & Deepika, R. (2018). Smart Home Automation and Security using Arduino. International Journal of Advanced Research in Computer and Communication Engineering, 7(6), 274-278. (ANPR) for Parking Systems. International Journal of Computer Applications, 134(1), 19-23.
[4]. Banerjee, S., Kumar, S., & Mittal, A. (2019). Smart Gate Automation with License Plate Recognition. International Journal of Engineering Research & Technology, 8(5), 670-673.
[5]. Sharma, A., Gupta, P., & Thakur, P. (2020). Home Security
[6]. Rehan, M., Khan, Z. A., & Siddiqui, A. A. (2016). Automatic Number Plate Recognition
[7]. Singh, R., Chauhan, M. S., & Pathak, S. (2021). Real-Time Face Recognition System Using OpenCV. International Journal of Advanced Science and Technology, 29(6), 7416-7425.
[8]. Kumar, N., Singh, S., & Rajput, D. (2018). Smart Lighting Control Using LDR. International Journal of Computer Science and Mobile Computing, 7(4), 124-130.
[9]. Rajan, R., Gupta, P., & Sharma, A. (2019). Gas Leakage and Fire Detection for Smart Homes. International Journal of Engineering and Advanced Technology, 8(6), 2596-2598.
[10]. Mohamed, S., Patel, J., & Shah, R. (2020). IoT-based Smart Home Security with Telegram Alerts. International Journal of Engineering Research and Technology, 9(9), 457-460.

