Research Inventy: International Journal of Engineering And Science
Vol.13, Issue 6 (June 2023), PP 88-95
Issn (e): 2278-4721, Issn (p):2319-6483, www.researchinventy.com

Design and Implementation of a AI-Powered Smart home system

Nguyen Cao Nguyen[1], Phan Thi Quynh Huong[2]
[1,2] *Vietnam Korea University of Information and Communications Technology – Danang University, Vietnam*
Corresponding Author: Nguyen Cao Nguyen

Abstract: *The utilization of embedded systems and IoT and AI technology is a novel approach that has revolutionized the conventional way of life, giving rise to a tech-savvy lifestyle. This has brought about changes in various domains such as smart cities, smart homes, pollution control, energy conservation, smart transportation, and smart industries, all of which are attributed to the advent of IoT, AI and embedded systems. Home and building automation is an advantageous and practical solution in this regard. This article focuses on the design and implementation of a smart home, showcasing its numerous useful applications, such as remote control, voice control, and door unlocking through image recognition methods.*

Keywords: *Smart home, Android application, AI technology, image recognition*

Date of Submission: 08-06-2023 Date of Acceptance: 22-06-2023

I. INTRODUCTION

In recent years, smart homes have appeared and are widely applied in life from hotels, luxury resorts, companies large and small to modern homes are equipped with smart control systems. Traditional smart homes are only controlled by applications, sensors or touch devices. But for users they always want to experience new technologies or applications more conveniently, and do not want boredom and when using any device. If users want to use any device, they can speak to the smart virtual assistant to ask for th services immediately, because the virtual assistant is everywhere in the house to easily serve everyone at all times.

In addition to the conveniences of the house, we need to ensure safety and also security. Currently, security is very limited and not common, so there are quite a few unauthorized home break-ins to carry out level thefts and threats, endangering users. Moreover, the issue of safety is also at the forefront, since recently there have been many explosions due to gas leaks mainly in homes using long-term gas stoves, which are not supported by a safety gas lock valve and do not have a system to warn.

The system has been thoroughly researched and incorporates highly secure and state-of-the-art technologies, including AI and IoT. With AI technology used to recognize faces to unlock the door for the owner, ensuring the accidental forgetting the key or other cases. IoT and Virtual Assistant were developed to provide convenience to users in managing furniture and controlling the use of electrical devices in the home with a network connection between lights and devices. Users must provide instructions for the system verbally and the system's sensors will recognize the user's voice to receive instructions and meet the user's needs.

This virtual assistant can be used flexibly in any location without taking up space and can also become a flexible help friend in the home.

The paper focuses on researching and implementing a smart home system that encompasses various functionalities, such as controlling devices through mobile applications, displaying calendars, news, and weather information, receiving voice commands via microphones, and utilizing photo processing applications to identify owners and facilitate door access. These modules represent the basic operations of the implemented smart home system.

In particular, the focus of this research is to address and improve upon certain issues encountered in the predecessor system as follows:

1) Renovate and upgrade the server that receives and returns the signals that control the devices.
2) Develop and develop utilities for virtual assistants such as: reviewing weather forecasts, turning on music, calendar of Vietnamese holidays, notifications when there are events,...
3) Performing an upgrade of the image processing module, improving the ability to recognize users with more powerful algorithms to increase accuracy with the camera.
4) Perform upgrades, interface changes, and other necessary extension functions.
5) Make more illegal break-in alerts by calling the landlord or registering an emergency number at the nearest police headquarters to promptly handle.

II. COMPONENTS DESCRIPTION

1.1 **Arduino Uno.**

The Arduino Uno R3, the latest version of the Arduino Uno, is a microcontroller board that utilizes a removable ATmega328 AVR microcontroller in a dual-inline-package (DIP) format. With 20 digital input/output pins available, it offers 6 pins for PWM output and 6 pins for analog input. Programming the board is made simple through the user-friendly Arduino computer program. Moreover, the Arduino enjoys a robust support community, making it an accessible platform for beginners interested in embedded electronics.

1.2 **Node mcu ESP8266**

Node mcu ESP 8266 is an electronic module mainly used in the IoT platform. Low-cost Wi-Fi is activated in it. It has GPIO, PWM, UART pins for communicating and controlling other external devices. Its hardware is based on the ESP 12 module. In the ESP 8266 node mcu board there is a connected CP2102 IC that provides USB to TTL functionality.

1.3 **Electric Lock Solenoid LY-03**

Solenoid Lock LY-03 electric latch lock comes with latch jig, which functions as a door lock using Solenoid for electric opening and closing jack, which is used a lot in smart homes or cabinets, electric doors. Locks using voltage 12 / 24VDC are usually closed with good quality, high durability.

1.4 **MQ-2 Fire warning**

The MQ-2 gas sensor uses a lower conductivity element SnO2 in clean air, when combustible gases exist, the sensor has a higher conductivity, the higher the concentration of flammable substances, the higher the conductivity of SnO2 will be and be correspondingly converted into an electrical signal level. MQ-2 gas sensor is a gas sensor with high sensitivity to LPG, Propane and Hydrogen, methane (CH4) and other flammable vapors, with low cost and suitable for various applications. The sensor outputs both analog and digital signals, the digital signal can adjust the alarm level by rheostat.

1.5 **Other devices**

- **Module Micro SD card:** is a micro SD card read/write module for Arduino using SPI communication, easy to use with SPI.h and SD.h libraries on Arduino IDE. Data can be read/written from micro SD. It allows implementing data logging projects, playing MP3 music.
- **Speaker – PAM8403 Sound Amplifier Circuit:** PAM8403 6W Hifi 2.0 Class D sound amplifier circuit (no volume) with a total power of 6W for 2 outputs and can be used with a 5VDC power source suitable for sound amplification applications that require compactness and low voltage (5V) use.
- **Webcam Logitech C270:** Take photos in 3 Megapixel real mode; 15 Megapixelsoftware. Record images in Full HD 1080, Video Call in HD 720.
- **12V Electronic Alarm:** Application modules make alarms, anti-theft circuits, alarms.

III. IMPLEMENTING THE SYSTEM

3.1. Implementing the smart home architecture

To build a smart home control system consisting of three main parts. The first part is the home control application, the second part is the server, the last part is the smart home model. To build the construction, I used Arduino Uno R3, module ESP8266, Python, NodeJS, JavaScripts, C language and weather datasets from openweathermap.org.

Figure 1 shows the smart home architecture.

3.1.1. Circuit diagram

Figure 2 and Figure 3 show the circuit diagram of the smart home and electromagnetic door lock system.

3.1.2. Building peripherals in a smart home

- Analysis of the requirements and functionality of the smart home:
    - Recognize the host's face to open the door with a magnetic lock.
    - Use the Android app to turn electrical devices on and off or communicate with Virtual Assistant.
    - Fully meet the basic requirements and functions of a smart home.
    - There are many utilities of Virtual Assistant such as: Hello, current time, google search, play music on Youtube, today's weather.
    - The fire alarm sensor will sound an alarm, making the owner more aware and flexible.
- Installation and execution steps:

- Install the ESP 8266 library into the Arduino IDE
- WiFi and Server connection
- Build LED lights, fans for each room Build Virtual Agent features
- Build gas and CO sensors to warn users, when there is a fire

Figure1. Smart home architecture

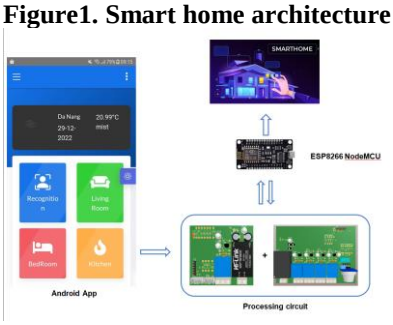


Figure2. Circuit diagram of the smart home

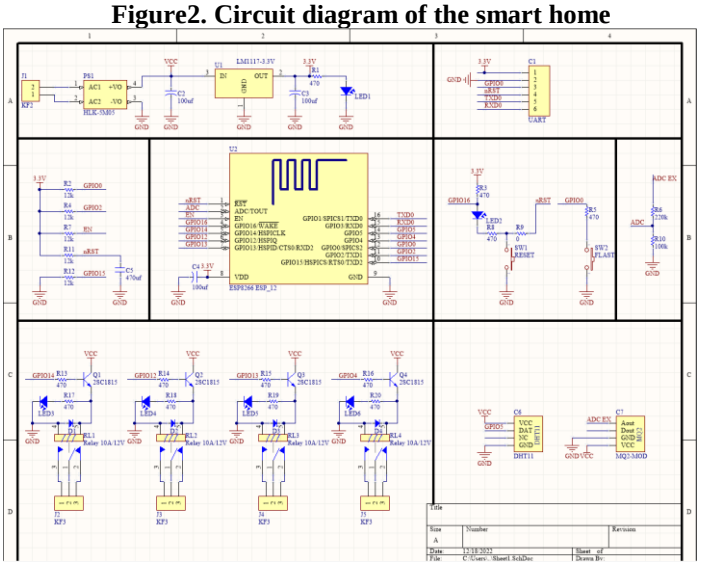


Figure3. Circuit diagram of electromagnetic door lock system

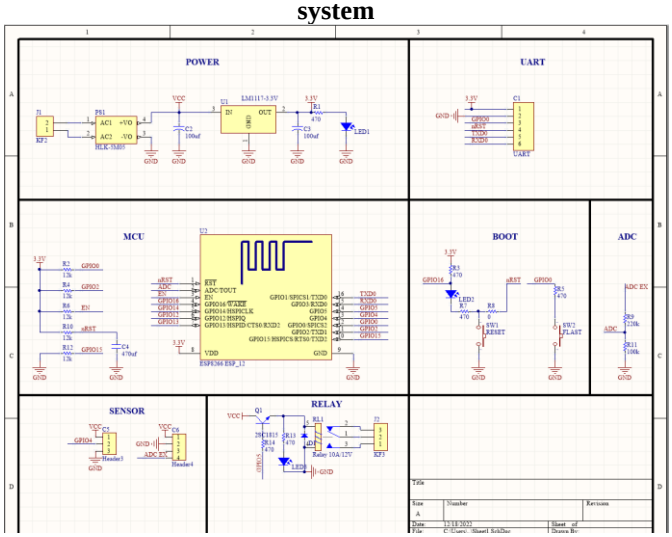
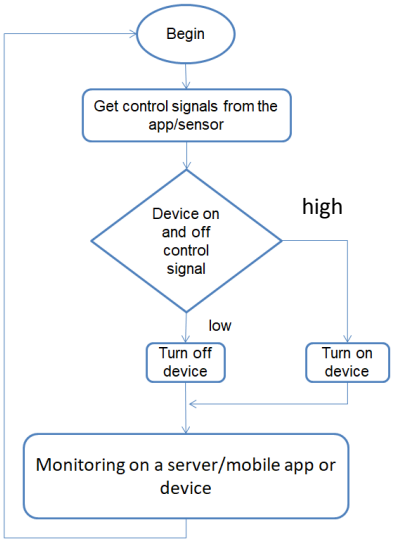


Figure4. Algorithm flowchart of a Smart Home



- Algorithm flowchart: Figure 4 shows the flowchart of the smart home.
- Operating principles:

When we execute the command through the Android application, then transfer to the server. The server will transmit the command to NodeMCU ESP8266. Here NodeMCU ESP8266 will process the data to stimulate the relay corresponding to the logical levels to turn the device on and off according to the user's on-off request and synchronize the state up to touch. In addition, the gas sensor will update the safety or danger status on the application and warn the user.

3.2. Implementing the smart home control android application

The use of the application to help users control devices more easily and intuitively, to better secure the home each user is registered an account and will be provided with an apk file to download the application to use.

The application will be changed and designed according to the number of devices as well as the functionality of the home that the customer desires. In addition, the application has built-in job notifications to help users not miss work or appointments.

- Analysis of application requirements and functionality
  - It is easier to use appliances in the house via wi-fi
  - User-friendly interface
  - Account login provided by the developer
  - Display hazard warnings for the house
- Installation steps
  - Install the Android Studio tool.
  - Implement firebase libraries.
  - GET-POST link from server

3.3. Implementing the virtual assitant

- Virtual Assistant requirements and functional analysis
  - Display date and time information, calendar notifications, reminders, news, and weather…
  - Google Assistant integration to be able to communicate with users.
  - Control electrical equipment by voice, integrated Vietnamese.
- Installation steps
  - Integrate devices like your phone's speakers, microphone, and screen for android apps.
  - Install Wifi and install sensors and equipment for the house system.
  - Download Python and add speech_recognition library.
  - Integrate module control with Virtual Assistant in C language.
  - Edit has a module displayed on the application that matches the user's requirements in the configuration file.js.
  - Install Google Assistant in the Assistant module file, set your account and select the language you use.
- Operating principles:

When called, the Virtual Assistant will listen to the request from the user to find related keywords then perform turning on / off the device or perform its utilities. When there is a danger due to a gas leak, the Virtual Assistant will notify the problem to the user and fix it immediately.

3.4. Implementing face recognition system architechture to open door

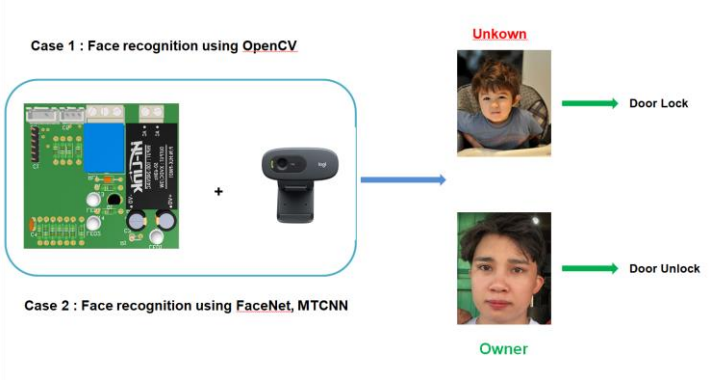
3.4.1. Collecting face data sets

The system collects images of facial data using the computer's own webcam, or from the phone device. The images collected need to ensure factors such as lighting conditions, different angles of the face, age,.. And the face should not have obstructions such as glasses, masks.

In addition, to ensure the accuracy of the system, for each user it is necessary to collect a certain number of photos not so little, and each photo contains only one face.

The photo dataset used in this project I used 100 photos of 1 family member. The exact number of photos for identification ranges from 30 to 50 photos per family member.

Figure5. Facial recognition system architecture to open doors



3.4.2. Select and label

We report 1:1 verification performance (cut between 224x224 from resized images with shorter edge = 256) on IJB-B for reference (ROC). The higher the better).

The model in the pre-train setup is trained in MS-Celeb-1M datasets and then fine-tuned on VGGFace2 datasets. The ResNet-50 model follows the inner architectural configuration and the SE-ResNet-50 model follows the inner configuration.

Ms-Celeb-1M is a dataset of more than 10 million images of nearly 100,000 celebrities around the world, including Vietnam, collected on the internet since 2016. It has now been discontinued by Microsoft.

3.4.3. 3.3.4 Detect the face for the input photo

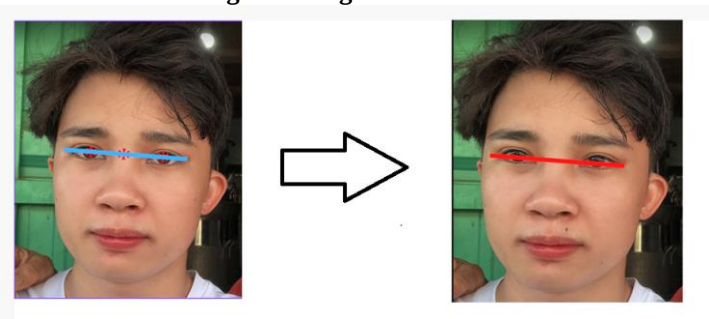
Input images can contain many faces, these faces may be not uniform in terms of light conditions, angles, expressions,... So I'm still going to use MTCNN as a facial detection tool because of its robustness.

3.4.4. 3.3.5 Align the face before identification

Facial alignment is quite important, which helps to improve some accuracy when introduced into identification. In this project, I use a 2D alignment method, which relies on facial landmarks (especially the eye area) to perform rotation, displacement, and adjust the proportion of the face to the same size.

Specifically, the algorithm takes the position of the left and right eyes from the MTCNN, calculates the middle point and proceeds to rotate the face so that the two eyes are on the horizontal line.

Figure6. Align the face before identification



3.4.5. Extract facial features from input photos

After detecting and aligning the face, the system will proceed to extract facial features using FaceNet. These characteristics will then be introduced into the "embeddings space" to compare with the dataset of existing features.

3.4.6. Facial recognition

In "embeddings space," faces in the form of 128-dimensional vectors are compared to datasets to find the most similar faces and label them. We used cosine similarity measurements to make comparisons. We use cosine similarity when we only care about the angle between two vectors and not the distance between them. Based on cosine similarity, we can calculate cosine distance using the formula: cosine_distance = 1 – cosine_similarity.

The input of the cosine function is the 128-dimensional vector array of the face in the faceless image that FaceNet has just calculated, and the dataset consists of arrays containing 128-dimensional vectors of pre-trained faces. Cosine distance will calculate in turn the distance between the input vector and the vectors in the dataset to find out the vector with the lowest value (shortest distance), corresponding to the highest similarity.

3.4.7. Identify "unknown" faces.

With an automatic door opening system in the house, it is possible to exclude cases where uninvited people come, or at times or hands, carry bulky carry and cannot open the door with their hands. Or we simply want to secure the house, only allowing relatives to come in and out and for strangers, these strange faces are in the end "unknown" faces.

"Unknown" faces are faces that are not in a pre-trained dataset. When photos appear these faces, if not identified and excluded, the chances that the system will mistakenly recognize the faces in the dataset and conduct the wrong opening is very high.

From cosine distance we can identify "unknown" faces at a certain rate based on the threshold. There is no standard rule for the selection of thresholds in facial recognition, for which this threshold value is selected through tests that yield the best results.

3.4.8. Comparison between the two cases of identification

Condition 1: Low light conditions, glasses and tilted face angle

Condition 2: bright enough conditions, no glasses, face angle

Table 1. Comparison between OpenCv and FaceNet

Test	Condition 1	Condition 2	OpenCV – Haar Cascade	FaceNet and MTCNN	Results
1	✓		✓		60%
2	✓			✓	83.65%
3		✓	✓		85-90%
4		✓		✓	95.87%

From the comparison table above I have the comments as follows

For condition 1: The results of OpenCV and Face-recoginition are not highly accurate, with adverse conditions, openCV's ability to enhance identification has not really convinced users for obstacle conditions and objective reasons.

In contrast to OpenCV, FaceNet is capable of recognizing with quite high accuracy and not to be confused with other faces.

For condition 2: The results of OpenCV and Face-recoginition are more accurate than condition 1, and for FaceNet the probability of accuracy in this condition is 10% higher than the trc condition.

3.4.9. Results

We have implemented various functions in our smart home, as depicted in the figures below.

Figure 7. Smart home model

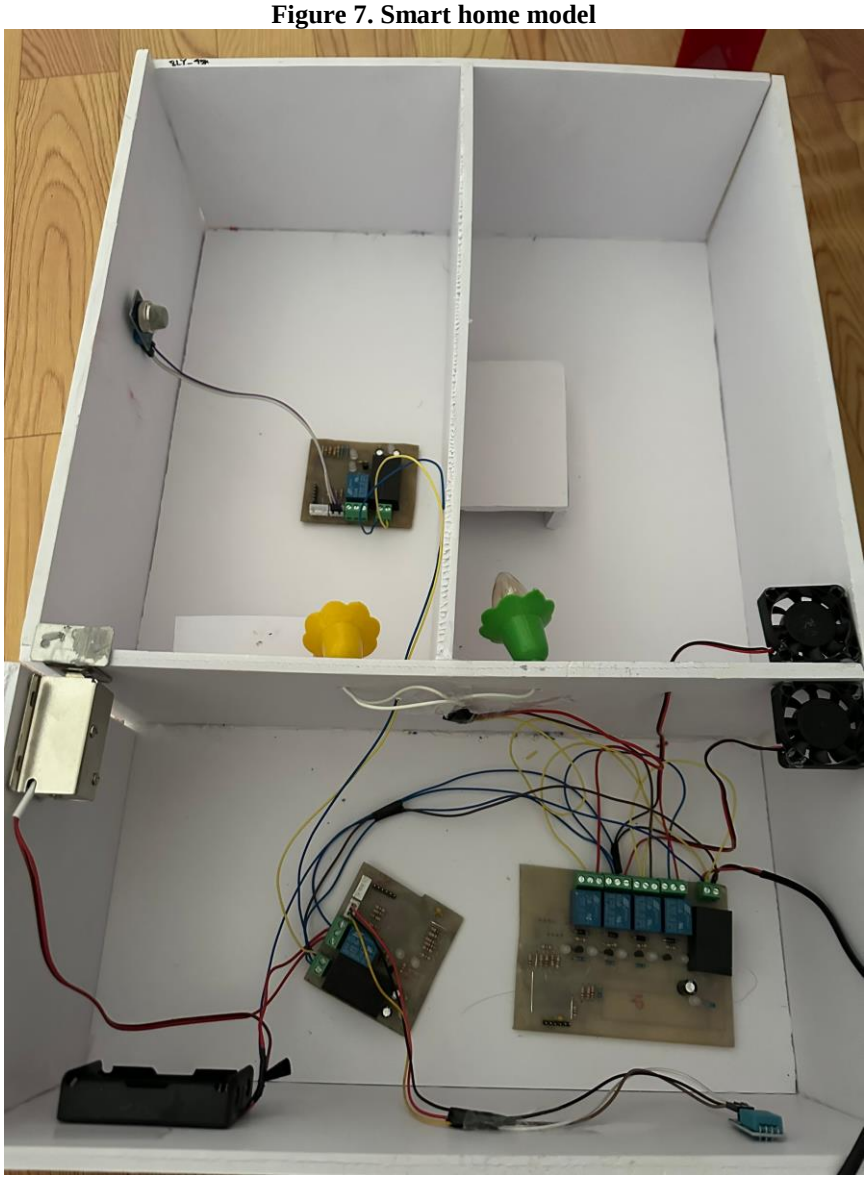


Figure 8. Smart Home Control Apps

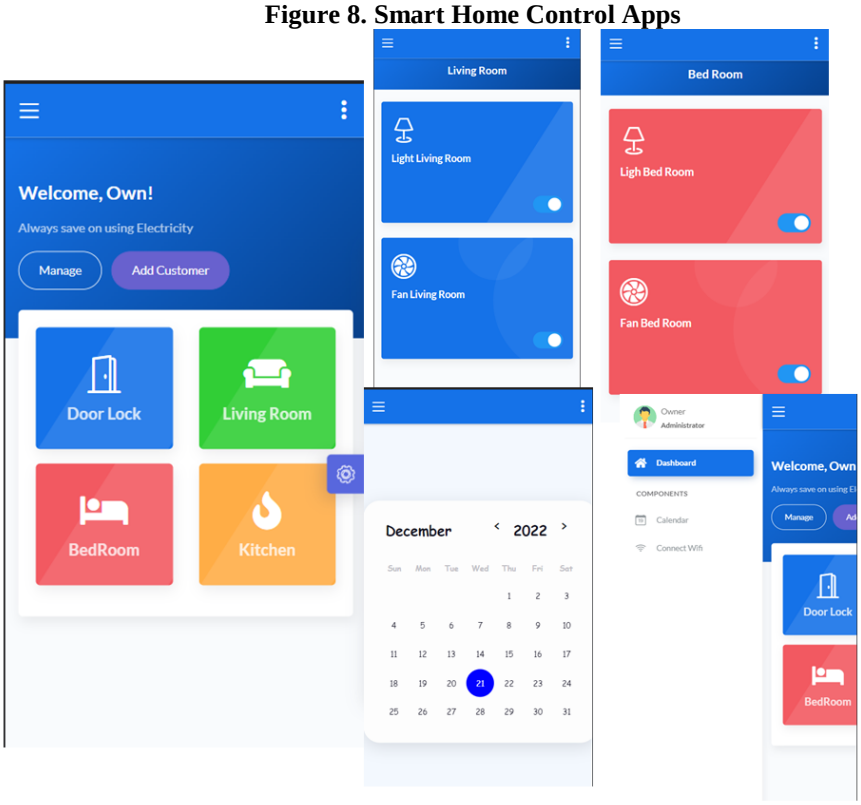


Figure 9. Samples of Virtual Assistant

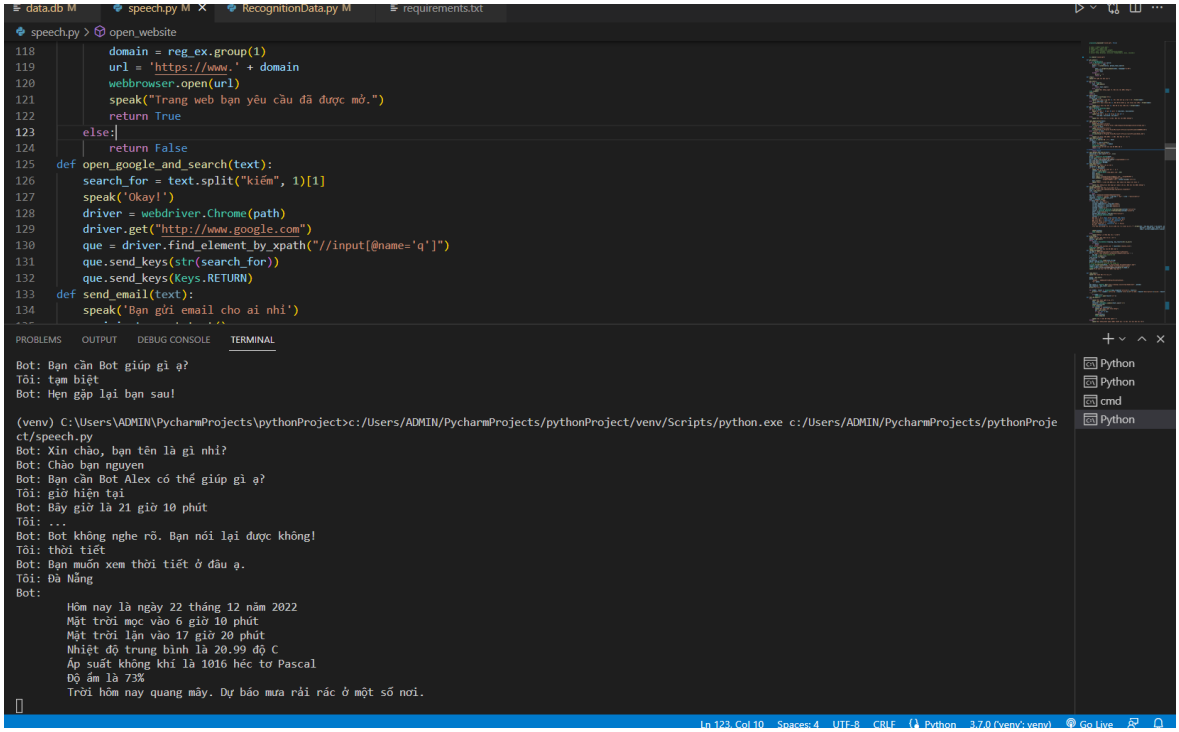


Figure 10. Result of OpenCV and Face-recognition – "Owner"

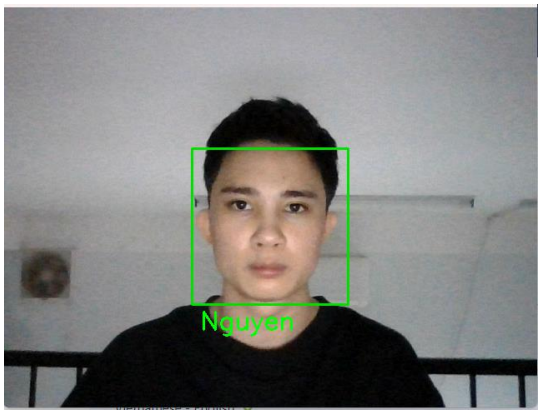
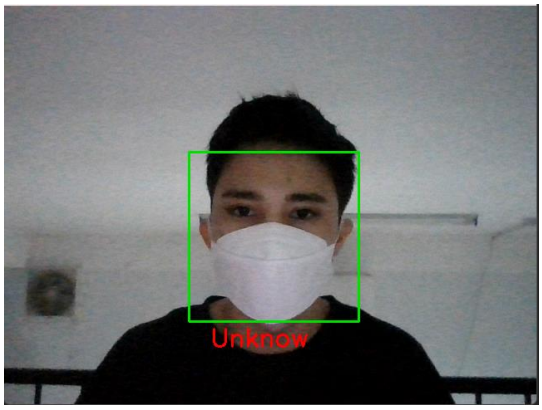


Figure 11. Result of OpenCV and Face-recognition – "Unknown"



IV. CONCLUSION

In general, the system has successfully achieved its basic functions and met the required criteria. The modules within the system operate harmoniously, and the devices respond to the user's commands through the Virtual Agent. The server of the smart home system remains operational 24/7, catering to users across different locations. However, there are still some challenges regarding recognition, particularly in varying light conditions and processing speed. The Face-recognition library is being optimized to enhance memory usage and CPU efficiency for quicker recognition. Furthermore, FaceNet, with its extensive academic library and multiple algorithms, requires additional processing threads and memory to improve recognition capabilities and reduce device boot-up time. Upon thorough analysis of the system, we have identified certain issues that need to be addressed. Firstly, improvements are required in the recognition capabilities and processing speed of the embedded computer. Instead of relying on communication with the server and awaiting a response, optimizing the internal communication within the house can enhance the transmission efficiency. By addressing these issues and implementing the necessary improvements, the overall performance and user experience of the smart home system can be further enhanced.

REFERENCES

[1]. G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. 2007. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst.
[2]. Bugsounet, "MMM-GoogleAssistant", https://github.com/bugsounet/MMM-GoogleAssistant
[3]. Malte Oeljeklaus, Frank Hoffmann, Torsten Bertram. 2018. A Fast Multi-Task CNN for Spatial Understanding of Traffic Scenes, Maui, 2018 21st International Conference on Intelligent Transportation Systems(ITSC), pp. 2825-2830.
[4]. S. Yang, P. Luo, C. C. Loy, and X. Tang. 2015. From facial parts responses to face detection: A deep learning approach, in IEEE International Conference on Computer Vision, pp. 3676-3684.
[5]. Florian Schroff, Dmitry Kalenichenko, James Philbin Google Inc, "FaceNet: A Unified Embedding for Face Recognition and Clustering"
[6]. Kamran, Sharif Amit. 2018. Efficient Yet Deep Convolutional Neural Networks for Semantic Segmentation, Yogyakarta, International Symposium on Advanced Intelligent Informatics (SAIN), pp. 123-130.
[7]. Nitin Agarwal, Build Cross-Platform Desktop Apps with Electron, https://www.wildnettechnologies.com/build-cross-platform-desktop-apps-with-electron/
[8]. T. Kurita. 2017. Improvement of learning for CNN with ReLU activation by sparse regularization , Anchorage, International Joint Conference on Neural Networks (IJCNN), pp. 2684-2691.
[9]. Vignesh Thakkar, Suman Tewary, Chandan Chakraborty. 2018. Batch Normalization in Convolutional Neural Networks Applications of Information Technology EAIT," Kolkata, 2018 Fifth International Conference on Emerging, pp. 1-5.
[10]. ByungSoo Ko, Han-Gyu Kim, Ho-Jin Choi. 2017. Controlled dropout: A different dropout for improving training speed on deep neural network, Banff, 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 972-977.
[11]. Géron, Aurélien. 2017. Hands-On Machine Learning with Scikit-Learn and TensorFlow:Concepts, Tools, and Techniques to Build Intelligent Systems, US: O'Reilly Media
[12]. Tsunenori Mine, Akira Fukuda. 2019. Intelligent Transport systems for everyone's mobility, Fukuoka, Japan: Springer.