VIRTUAL HOME ASSISTANT USING FACE RECOGNITION

MUHAMMAD AFIFF EIZLAN BIN NORAZLAN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

VIRTUAL HOME ASSISTANT WITH FACE RECOGNITION

MUHAMMAD AFIFF EIZLAN BIN NORAZLAN

This report is submitted in partial fulfilment of the requirements for the

Bachelor of Computer Science (Networking) with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I hereby declare that this project report entitled

VIRTUAL HOME ASSISTANT WITH FACE RECOGNITION

is written by me and is my own effort and that no part has been plagiarized

without citations.

STUDENT: **MUHAMMAD AFIFF EIZLAN BIN NORAZLAN** Date : 21-06-2024

I hereby declare that I have read this project report and found

this project report is sufficient in term of the scope and quality for the award of

Bachelor of [Computer Science (Software Development)] with Honours.

SUPERVISOR : DR. ZAHEERA ZAINAL ABIDIN
Pensyarah Kanan
Fakulti Teknologi Maklumat Dan Komunikasi (FTMK)
Universiti Teknikal Malaysia Melaka (UTeM)
Date : 21-06-2024

([DR. ZAHEERA ZAINAL ABIDIN**])**

DEDICATION

I would like to extend my heartfelt dedication to my parents, Norazlan bin Marjo and Siti Hajar binti Mohd Bakir, for their unwavering love and support. Your guidance and encouragement have been instrumental in shaping my academic journey.

I am immensely grateful to Dr. Zaheera Zainal Abidin for his invaluable guidance and mentorship as my supervisor. Your expertise and insights have been instrumental in shaping the success of this project.

Special thanks to my dear friend, Mohd Amirul Akmal bin Mohd Tarmizi, for his constant motivation and encouragement. Your belief in my abilities has pushed me to excel and overcome challenges.

To my friends, Nur Salihin, Nazrin Hanif and Muhammad Aiman, thank you for your unwavering support and assistance throughout this project. Your collaboration has made this journey more enjoyable and rewarding.

Lastly, I dedicate this achievement to myself. It is a testament to my hard work, determination, and perseverance. I am proud of the growth and progress I have made throughout this journey.

To all those who have played a role in my academic and personal development, thank you for being a part of my journey. Your presence and support have been invaluable.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude and appreciation to the following individuals who have played a significant role in the successful competition of my final year project.

First and foremost, I would like to thank my parent, Norazlan bin Marjo and Siti Hajar binti Mohd Bakir, for their unconditional love, unwavering support, and constant encouragement throughout my academic journey. Your belief in me has been a driving force behind my achievement, and I am truly grateful for everything you have done for me.

I also like to extend my heartfelt appreciation to my supervisor, Dr. Zaheera Zainal Abidin, for his guidance, expertise, and valuable insights. Your mentorship and encouragement have been instrumental in shaping the direction and quality of my project. I am truly grateful for the knowledge and skills I have gained under your supervision.

Special thanks to my dear friend, Mohd Amirul Akmal bin Mohd Tarmizi, for his continuous encouragement and motivation. Your words of support have been a source of inspiration for me throughout this project. I am grateful for your belief in my abilities and the positive impact you have had on my journey.

I would also like to express my gratitude to my friends, Nur Salihin, Nazrin Hanif and Muhammad Aiman, for their assistance, collaboration and unwavering support. Your contributions and dedication have been invaluable, and I am grateful for the moments we shared and the teamwork we exhibited.

Finally, I would like to acknowledge and appreciate myself for the hard work, determination, and resilience I have demonstrated throughout this project. This journey has challenged me, allowed me to grow, and empowered me to achieve my goals.

To everyone mentioned above, thank you for being a part of my journey and for your contributions to my success. Your support and presence have made a significant impact, and I am truly grateful for all that you have done.

ABSTRACT

In recent years, improvement in artificial intelligence and computer vision have sets out the path for the innovative applications in home automations and personal assistance. The goal of this project is to develop the virtual home assistant that integrates with face recognition for the security improvement and user interaction in smart home environments. Virtual home assistant that integrates with face recognition will identify and authenticate users, providing a smooth, hands-free experience that can be customised to each user's preferences.

The primary features of virtual home assistant including device management, identification of unauthorized access attempt and responses to the users. By using the machine learning models, the system will improve the accuracy of the face recognition to the facial changing over the times.

This project involves designing, implementing, and evaluating the virtual home assistant with focus on user-centric features and robust security mechanism. The result show that using face recognition integrate with virtual home assistant will not only makes user interactions more efficient, but it also greatly improves the security for the current smart homes.

Table Of Contents

List Of Tables

PAGE

List Of Figures

PAGE

Chapter 1

1.1 Introduction

Nowadays there are many virtual assistants where they are developed by various developers in the world of technology. There is also smart home that can integrate with the hardware that include in our home such as light and fan. Most of this technology doesn't have security which is there is no specific user that can use this technology so how the assistant can verify user identities and grant appropriate access to smart home devices and assistant functions.

This project will integrate the virtual home assistant with the face recognition for the better security. When the user wants to use the smart home, the device will use the camera for the face recognition then if the system detect that the user is authorized it will give the permission to access the function of smart home and also the function of the virtual assistant that integrate with the smart home system. Normally the virtual assistant that available in our live can use the function for all of the users. Based on the research, the main problem of the smart home system nowadays is the security issue. Unauthorized users can potentially access and control the functions of smart home devices due to poor authentication mechanisms, communication protocols, or the lack of a secure channel between the user and the device (20 Jun 2023-ARPHA Conference Abstracts). Based on another research (Sirisha, Uppuluri., G., Lakshmeeswari. (2022)), The unauthorized user also may be able to gain access to smart home device and manipulate the functions. For example, the attacker could use these vulnerabilities to turn off light or even steal personal information. Based on this research also state that the problem on smart home is lack of secure user authentication and key agreement for IoT device access control in smart home communication. to make

the smart home system more secure to the user, so this project will add the face recognition to the smart home system.

When the system detects the authorized user it will allow the user to use all the function of the system. It involve the identification and verification of individual based on their facial features (14 Apr 2023-Sinkron : jurnal dan penelitian teknik informatika). Face recognition adds an extra layer of security to smart homes by allowing access only to authorized individuals, preventing unauthorized entry (Gesha Warilotte Erwinda-26 Jan 2023).Face recognition can be integrated with other smart home systems such as surveillance cameras and send the notification to the telegram app if it detect the unauthorized user want to access to the smart home system. This system is also equipped with a virtual assistant that can interact with the user. This system will provide with voice user interface that configure in the Raspberry Pi. Voice User Interface (VUI) is an interface that allows users to interact with a system using voice commands. It enables users to control and interact with devices or applications through speech recognition and synthesis technology (P Srinivas-30 Apr 2020).

1.2 Problem Statement

Nowadays, existing virtual home assistant cannot identify between the individual that using the home assistant, and it don't provide the user-specific modification. This project aims to develop the home virtual assistant that will integrate with the face recognition for the users verifying and provide personalized services.

No	Problem Statement
PS1	The main problem of this virtual home assistant is it can be access by all the users and it make that the personal and important data can be access to all the users. This project will develop virtual assistant that integrate with face recognition for the security method to verify the users that want to access to the system.

Table 1.1 Summary of Problem Statement

1.3 Project Question

In the context of this project, the question has been created and summarized in the table 2.1 with the reference from the problem statement. These are the questions that come up and need to have their answers.

PS	PQ	Project Questions
PS1	PQ1	Can face recognition be used for the improvement the user authentication and security?
	PQ2	What is critical characteristic that needed for the create a flexible virtual personal assistant that can give the best user privacy and security?

	PQ3	Can virtual home assistant's voice command features be improved for the accurately and successfully manage the devices and provide user-specific services?

Table 1.2 Summary of Project Question

1.4 Project Objectives

By referring the problem statement and project question above, the project goals will be determined at the table 1.3.

PS	PQ	Project Objectives
PS1	PQ1	To analyse the user authentication for the secure smart home
	PQ2	To develop a secure virtual assistant for smart homes using face recognition
	PQ3	To evaluate the function of the smart home and virtual assistant by using the user's voice

Table 1.3 Summary of Project Objectives

1.5 Project Scope

This project will create, develop and deploy an innovative virtual home assistant that use face recognition to offer the user's information safe, and for the interaction in the smart home devices. This will include improving the virtual assistant security through face recognition and replies to the services that required from customers. To ensure the user-friendly and secure smart home experience, the assistant will include the voice command capabilities that enable simple and effective control over the smart home features.

1.6 Project Contribution

The purpose of this project is to make a virtual home assistant that integrate with face recognition. The study aims to develop the system that can enhanced the security and can the system that convenience to the users. This system doen't need the physical keys or codes to unlock the function of the system, offering a seamless user experience. By choosing the suitable face recognition algorithm, it will determine the possibilities of face recognition in the virtual home assistant environment. The main contribution is assessing the viability and effectiveness of facial recognition in virtual home assistant system, taking into consideration of the accuracy of the detection and the security enhancement.

1.7 Report Organization

This report contains 7 chapter to be discussed. This section will summarizes the content of the report which start from chapter 1 called introduction until Chapter 7 which is conclusions.

Chapter 1: Introductions – this chapter will discuss about introduction, project background, project problem, project questions, project objective, project scope, project contributions and report organization.

Chapter 2: Literature Review – this chapter will discuss the literature review and the detail of the previous project that related to this project. It will contain the related works, critical review of the current problem and report organization.

Chapter 3: Methodology – This chapter will contain the project methodology, details of its phases of the development, activities done and project milestones.

Chapter 4: Design – this chapter will discuss about the detailed design of the proposed tool with the diagram. It also discusses the hardware and software requirements.

Chapter 5: Implementation – this chapter will discuss about the configuration of the hardware and software along with the coding of multiple IoT components.

Chapter 6: Testing – This chapter will provide the method to test the project and show the result that get from the testing phases. The testing is required the final version of the prototype.

Chapter 7: Conclusion – The overall conclusion of the project is discussed in this chapter. It will contain the project summarization, project contribution and project limitations, along with future works that be done to improve the further project.

1.8 Conclusion

In conclusion, this chapter provide the explanation the goals of this project, which is included introduction of the project, the problem statement, project objectives, scope of the project, and project contribution. The overview of each chapter also provided in this chapter. The literature review of the project along with the any previous work and the gap of study will be discuss in the following chapter.

Chapter 2

Literature Review

2.1 Introduction

This chapter focuses on the main ideas and theories of the project virtual home assistant using face recognition. In this chapter will examine several projects in the last 5 years related to the project to be developed. The concept, technique of project, problem statement and gap of study that involve in this project and the ways to solve the problem will be discuss at this chapter. Articles, case studies, and journals from the past 5 years are the main sources of knowledge. These resources were selected by the similarities project scopes.

2.1.1 Virtual Home Assistant

Home virtual assistant is the software that can be done in various task and responds to the user commands in the home environment. There are many ways for the user to give the command to the virtual home assistant, such as human speech, respond via synthesized voices and convert voice to text. It can control the home automation devices, provides weather updates and interacts with users by sign language.

The main objective for the virtual home assistants is to automate task and make them more convenient for the user. This can reduce the user's workload and paves the ways for the system to eventually manage additional jobs on its own. Wake-up word, hand gestures or voice instructions can all be used to activate it. To accomplish task with the speech, text, and image inputs, home virtual assistants make use of technologies including natural language processing, convolutional neural networks,

and sign language processing. Virtual assistant accuracy and responsive are enhanced by these technologies. Technological developments in artificial intelligence, machine learning, neural networks, and Internet of Things integration could potentially expand the functionalities of virtual assistants at home. Further versions of virtual assistant systems may result from the merging of these technologies.

The virtual home assistant also knows as the smart voice assistant that connect to the internet and can perform various task based on user instructions. They can extract information online, manage resources, notify users about incoming messages and reminders, and more. This virtual assistant can take instruction in text or voice, and it also have the feature wake-up call to activate the listener. They create by using the compiler such as phyton. They also created by using performance intelligence and machine learning.

They use machine learning algorithms and natural language processing technique to recognize and interpret user command. They also rely on wake-up call or specific word to activate the assistant and start processing command. Ms. Shruti Sarwate 2 August 2022

2.1.1.1 Voice Recognition

Voice recognition is a biometric method that makes use of an individual's distinctive voice characteristics to facilitate speech-based technological communication. It entails translating spoken language into signals that computers can comprehend, enabling users to carry out manual tasks like making notes and asking questions. Voice recognition has become safer, more convenient, more secure with recent advances in computing technology, yet issues like linguistic diversity and background noise persist. Yudi Aryatama Fonggi, Tio Oktavianus 31 May 2021.

On the other paper, voice recognition is a type of computer analysis system that uses human speech as input to process data. It can identify and distinguish between different human voices. The voice recognition can be applied in many fields such as telephony, voice assistant, forensic and access controls.

The process by which a computer recognises a voice varies depending on the system; for example, a genetic algorithm can identify a word spoken by 25 different users and correctly identify 100% of the words, while the Fourier method can be applied by feeding the computer some input data to identify a distinct voice signal in a numerical pattern, which can then be used by the computer system to learn and recognise a human voice. Another method utilised in this research computes a signal pattern using magnitude analysis, which can serve as a foundational technique for a voice recognition system and be expanded upon with other systems or technologies for different applications. Hassan Falah Fakhruldeen, Heba Abdul-Jaleel Al-Asady 1 August 2023.

2.1.1.2 Face Recognition

Face recognition is a method to identify and verify the person identity based on the features on their face. It is the most popular biometric recognition that has gained attention on the research. It will analyse various factor such as background, head posture, brightness and face shape which can affect the accuracy of identification. Titiwat Kuarkamphun, Chiabwoot Ratanavilisagul 21 December 2022.

This recognition does not require physical interaction from the user on the sensor. The process for the face recognition to match the user face is it will use Singular Value Decomposition (SDV) algorithm. The picture will split into component element to extract the most significant aspects.

Then the image will classification by using Hidden Markov Model algorithm. This algorithm will be divides the images into the area and assign a state to each area. The average height of the region that a certain state cover determines the possibility of access across states.

To train the system it requires almost 100 sample images which is half of the sample is to train and the rest of the sample is to test the system accuracy. This combine if the SVD and HMM algorithm shown that the improvement in the recognition rate which is from 94% to the 96%. Nanthia, Suthana 1 January 2023.

Facial feature analysis and processing are a key component of face recognition, which is used for identity, security, and authentication among others uses. It will be detecting faces by using the camera and it will be process with the specific algorithm to identify and match faces accurately using the distance method. Then the system will store the information in a databased and compares the camera captured dimensions of

a student's face with the database's stored images. S. Bazith and Preety Savant 30 May 2023.

2.1.1.3 Gesture Recognition

Physical motions made with the hands, fingers, arms, head, face, or torso are called gestures. Gestures are a significant means of communication in many important situations. They enable computers to interpret the performer's intentions and, as a result, these devices can be given precise commands.

Gesture data can be captured using computer vision or wearable technology. Specialised wearable technologies, such data gloves, can collect gesture data that can identify precise finger curvature and hand and arm spatial position information. Realising gesture recognition involves the examination of user modelling and gesture films.

Wearable technology can achieve improved recognition accuracy by collecting more precise data about hand movements. Nevertheless, this approach requires specific technology that is costly and inconvenient to use daily. On the other hand, computer vision uses just a camera and no further hardware to record user hand gesture movements. Yuanyuan Shi, Yunan Li, Xiaolong Fu, M.I.A.O. Kaibin, M.I.A.O. Qiguang 1 Jun 2021.

On the other research, the advantages and disadvantages of hand gesture recognition technologies vary depending on the platforms in which they are used. Realistic and efficient real-time hand gesture recognition is currently hindered by multiple challenges faced during foreground separation from the backdrop. The foreground represents the hand that must be identified. The most frequent issues are

shifting image brightness, such as the background and hand skin pixel colour in vision-based systems, and heavy, costly equipment in glove- and depth-enabled systems.

Hand gestures are a type of body language in which information is communicated by the location and form of the fingers and palm centre. The gesture consists of both static and dynamic hand gestures. Static hand gestures are solely dependent on the shape of the hand, whereas dynamic hand gestures consist of a sequence of hand motions. Because gestures vary widely across cultures, various people will characterise them in different ways. Dynamic hand signals rely on the movement of the hands to transmit the meaning, whereas static hand gestures rely on the shape of the hand gesture. Real-time hand gesture detection is the capacity to recognise hand motions quickly and without delay. Real-time and non-real-time hand gestures differ in terms of processing speed, picture processing methods, allowable result delay, and recognition algorithms. Fahmid Al Farid, Noramiza Hashim, Junaidi Abdullah 26 May 2022.

Template	Biometric	Researcher
Voice Recognition	Voice pattern	- Yudi Aryatama Fonggi, Tio Oktavianus 31 May 2021. - Hassan Falah Fakhruldeen, Heba Abdul-Jaleel Al-Asady 1 August 2023
Face Recognition	Head posture, brightness of skin, face shape	- Titiwat Kuarkamphun, Chiabwoot Ratanavilisagul (2022) - S. Bazith and Preety Savant (2023) - Nanthia, Suthana (2023)
Gesture Recognition	Finger, palm, hand	- Yuanyuan Shi, Yunan Li, Xiaolong Fu, M.I.A.O. Kaibin, M.I.A.O. Qiguang 1 Jun 2021. - Fahmid Al Farid, Noramiza Hashim, Junaidi Abdullah 26 May 2022.

Table 2.1: Different Of the Recognition

2.2 Virtual Home Assistant Using Face Recognition

Based on the project that conduct from Ondra, Eka, Putra., Retno, Devita on 2 April 2023, the project that has been build is face recognition that connect to the IoT devices. The IoT devices has been configured to the system that can detect the authorized user. If the system detect that the user is unauthorized user, it will send the email to the owner in the real time. The system has several stages that reveal in the methodology as shown in figure 2.1.

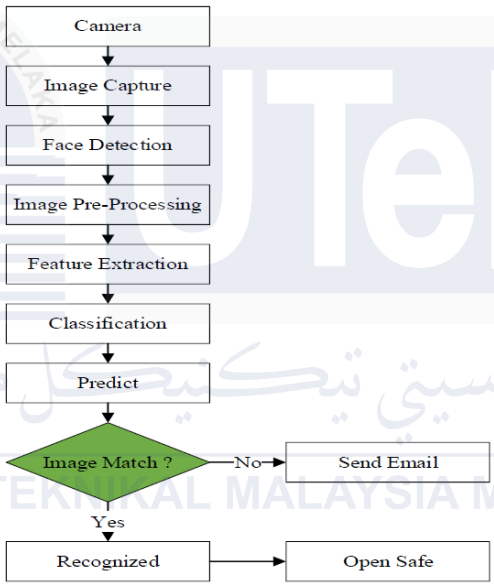

Figure 2.1: Stage In Project

Based on the figure 2.1, the way to operate the face recognition is by using the camera. Then the camera will capture the image of the person and required the name of the user and the data that has been receive will stored and then it will detect the face if it is the authorized user by using the system that has been configured by the deep learning. The way to detect the face is by determine the size and location of the face in the input images. Then the system will go to the classification process and predict the result by using the classification result. Sorting objects into the proper classes is

the process of classification. Following the classification of the data, facial recognition is used. If the face that has been capture is detect the user is the authorized user, it will open safely but if it is the unauthorized user the system will send the email message to tell the user that there has the unauthorized user try to use the system.

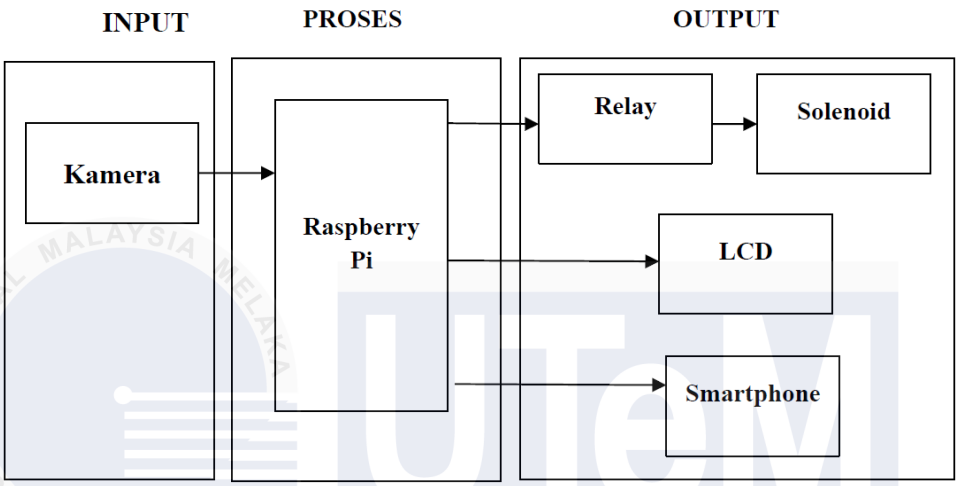

Figure 2.2: Diagram For Safety Camera

From the figure 2.2 show the way of the safety camera works in the diagram. The camera will attach to The Raspberry Pi where the configuration of the system will be stored in this device. Then in the Raspberry Pi it will process the image to determine if it is the authorized user or not. If it detects the intruder, it will send the notification to the smartphone and the lcd to tell the user that it detects the intruder.

Figure 2.3: Camera Detect the Face

Figure 2.4: Notification Of the Intruder

In this paper, the algorithm that use for this face recognition is eigenfaces method, which involves Principal Component Analysis (PCA) algorithms. This algorithm will be cropping face photos to smaller sizes without sacrificing crucial characteristics. It will be utilised in facial recognition systems for feature extraction. It functions by cropping facial photos to a smaller size while maintaining the critical details required for precise identification. When it comes to face identification, principal component analysis (PCA) examines how different face photos in a dataset vary and finds the most important patterns or elements that best capture the data. Without sacrificing crucial details, these elements successfully depict the faces' main features, such as the forms of the mouth, nose, and eyes.

By converting the initial high-dimensional face photos into a lower-dimensional space delineated by the principal components, PCA is able to achieve dimensionality reduction. The directions with the largest variation in the data are represented by these orthogonal vectors, which are called main components. The most pertinent information is efficiently captured using PCA while less significant aspects are discarded by projecting the facial images onto these primary components.

This makes it possible to store faces of people in the database effectively without experiencing a large loss of data. The technique creates a set of eigenfaces that represent the differences in facial features among individuals by applying PCA to

extract significant features from the face images. The system is able to precisely match detected faces with the stored profiles by using these eigenfaces as a foundation for comparison during the face recognition process. All things considered, PCA improves the system's face recognition performance by offering a reliable technique for feature extraction and comparison.

In other study it aims to create the system that using the face identification by collecting data with the vision framework based on a Raspberry Pi. The Raspberry Pi vision system will take the picture, which is subsequently sent to cloud storage for matching. After that, highlights from the confrontation are extracted using deep learning-based new engineering, and they are compared to a database of known individuals who have been granted access to the smart houses. The smart gate will unlock and grant entrance to the authorised person if the face it detects matches the faces in the database of known people. However, the smart gate will stay shut and entrance will not be allowed if the recognised face does not match any authorised users in the system. The diagram of the study is show in the figure 2.5.

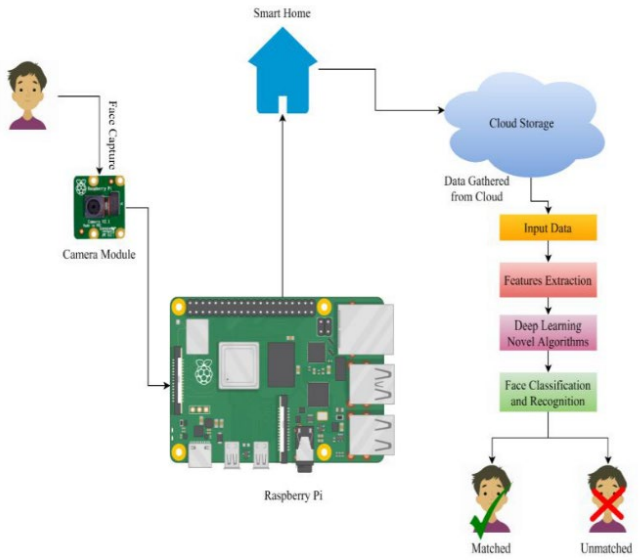

Figure 2.5: Diagram Of the Device Setup

To enhance classification performance, the research attempts to develop a face recognition architecture that integrates the SVMBoosted algorithm with Convolutional Neural Network (CNN). The convolutional layer will extract features from the input image from the local database, and the max pooling layer will come next. To identify each face, the extracted features from the Convolutional Layer are used to analyse its texture and compare it to the reference or expected data, also known as ground truth. For the purpose of increasing the system's resilience, the retrieved values will be kept in a CSV file. The SVM-Boosted computation will identify the right confront locked after extracting the CSV record.

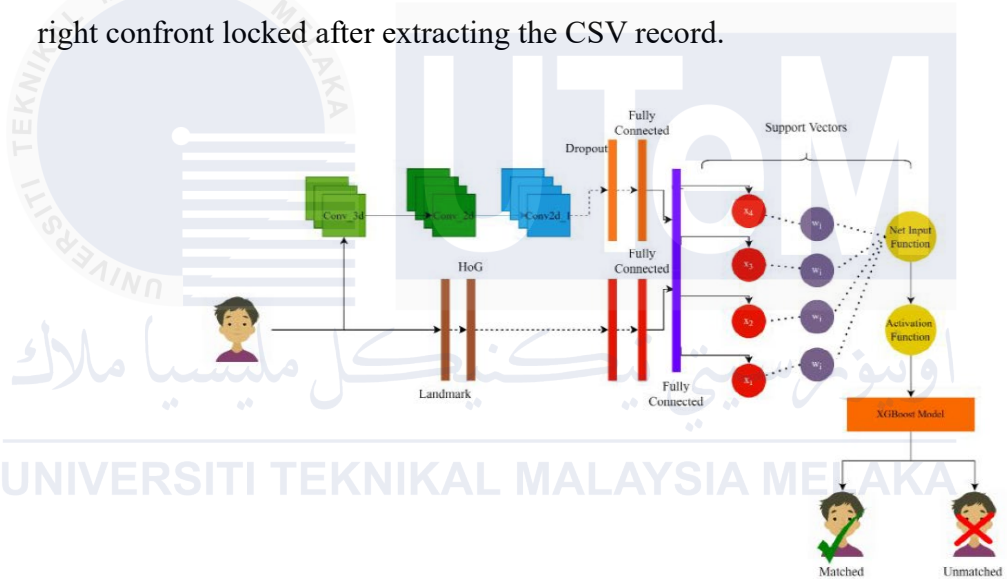

Figure 2.6: Proposed Model Architecture

The purpose of texture classification, a challenging task in pattern recognition, is to distinguish between various textures. Given the number of advanced classifiers available, one of the key issues is determining important characteristics that may be recovered from the textured image. Both tactile and optical textures can be experienced; tactile textures are felt by contact with the surface, whilst optical textures are related to the shape and content of the image.

Different functions can be employed by machine learning models to analyse data in a given area. Raw data must first be transformed into a format that the model can understand and use, which typically calls for manual labour, before utilising machine learning techniques. A correlation matrix was employed by the researchers in this study to assess the link between various factors. Like a covariance matrix, a correlation matrix indicates the strength of the linear relationship between variables. A correlation matrix is used in this study to assess the link between several variables. The strength of the linear link between variables is summarised in a correlation matrix, which is a covariance matrix. A summary metric that shows the direction and intensity of a linear relationship between two quantitative variables is called the correlation coefficient, or "r." Perfect negative correlation is represented by a value of -1, perfect positive correlation by a value of +1, and no correlation is represented by a value of 0.

On the other paper that using the CNN algorithm, webcam will send the UID data to the device that use to open the solenoid door lock. After that the device will send the data to the NodeMCU and it will send the data is entered into the Database and will display in the website. Figure 2.7 below show that the system block diagram.

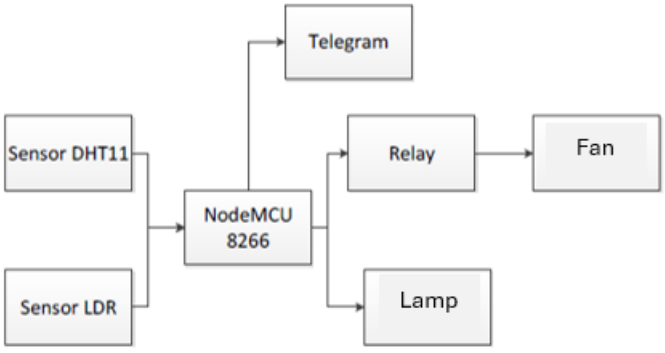

Figure 2.7: System Block Diagram

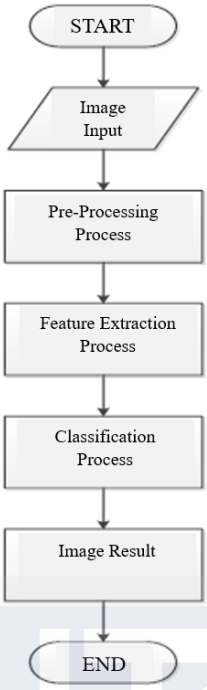

Figure 2.8: Flowchart For CNN Method

Based on the figure 2.8 show that the process of the CNN algorithm that use for the face recognition. First, using a camera or other image-capturing equipment, the process begins with taking pictures of people's faces. After that, these photos undergo pre-processing, which includes grayscale conversion and standard size scaling. The input photos are standardised in this stage so they can be processed further. Feature extraction comes next after pre-processing. To identify the person in the picture, the CNN analyses the image in this stage and extracts any pertinent aspects or attributes. The CNN extracts hierarchical features at various levels of abstraction by processing the picture data through numerous layers of pooling, convolution, and activation functions. The key details about the face features are then captured in a numerical format by converting the retrieved features into a feature vector or matrix representation. The categorization stage then uses this feature representation as an input.

The retrieved characteristics are sent into the CNN for training and learning during the classification stage. Based on the extracted features and their patterns, CNN gains the ability to distinguish between various individuals. The CNN modifies its settings throughout training to maximise recognition accuracy. The smart home security system can employ CNN for real-time face recognition after it has been trained. To identify a person, a CNN examines a fresh face image, extracts feature from it, and compares those traits with previously learnt features. The system identifies the user and provides access if the retrieved features match a template that has been saved.

First, the system will detect faces, then the capture images of the user will compare with the data that has been previously trained in the model-cnn-facerecognition.h5 file, otherwise the users must register their faces first into the website while registering e-KTP. If successfully recognized then the buzzer will light up, LCD displays the message as well as the Door Lock Solenoid will opens and will return to facial recognition. If the Webcam experience an error for recognize faces such as the Webcam is having problems, then the user can scan their existing e-KTP previously registered to be able to open the Solenoid. For this system only the admin can manage the website for the login, show the history of the access, add the data and access to the security system such as the webcam and the door lock.

On the other study, with the use of deep learning, a smart home may now offer a high degree of safety and medical services, including the capacity to record and identify human activities and faces using a depth camera. With daily and continuous monitoring of human activity, attempts are being made to optimise these systems to improve the quality of life in general and the quality of life in particular for the elderly. Within the home, the system continuously records every day, gaining experience and

knowledge in identifying events and behaviours. This is important information that improves the system's functionality and, as a result, raises the standard of services offered. A system like this can record life in real time by using depth imaging to identify human activities. A machine that uses deep machine learning and neural networks can identify, categorise, and align facial photographs. Given a 2D image, the machine can produce an aligned 3D image. facial recognition is achieved in this procedure by classifying and aligning the 2D facial image. A person's distinctive qualities are recognised as a collection of vector values that provide the person's distinct identity. Without the need for human assistance, the algorithm uses a picture or video to identify a person. Face identification in commonplace photographs captured in real-world scenarios is an innovative and ground-breaking computational idea. Even while the machine attempts to mimic the human brain and vision, it is still far from perfect and has a high error rate when it comes to facial recognition. Even with the significant progress made in recent decades, there is still room for improvement in the system's ability to recognise two-dimensional faces captured in strictly regulated settings. The 2D picture is used for recognition. The aligned 3D image is used to classify this using neural networks. A collection of features are included in the identification process for the 2D image. An image contains a set of 2D key points that are found in a genuine system application. The 3D aligned face image can be obtained by first creating a 3D form from the 2D face image. A second set of important spots in the image can be detected during the conversion process to identify a portion of the face. The creation of the whole 2D image may result from this set. Actually, a set of anchors is extracted from the two-dimensional picture from where the picture is created. The DNN is composed of a converging layer and a layer of maximum concentration. The convergent layer, when coupled to each of its adjacent

layers, extracts the three-dimensionally aligned points of the face in each of its functions. Every group of linked layers is set up to output the feature face vector, which is the correlations between the three-dimensionally aligned points of the face. DNN then sorts the three-dimensional image using these vectors. The feature vectors identify each feature in the 3D image and normalise it to a predetermined range. Every pixel is given a three-dimensional aligned point using defined filters. Using the set of data and vectors it has available at any given time, the machine learns to define the filters on its own. A database of photos can be used to identify a 2D facial image. A face database is used to store face images. Each group of photographs in the database corresponds to a certain face. Face ID is a set of features. The sorting of each face ID obtained from a 2D image is done by assigning it a unique value throughout the execution of a DNN between its neighbouring layers.

Deep Neural Networks (DNN) a group of algorithms that make advantage of the backpropagation method to efficiently train artificial neural networks. In order to identify the loads of neurons and train the network, the backpropagation method repeatedly feeds back newly updated input data until the desired output results are obtained. DNNs are made to mimic how the human brain picks up knowledge through experiences and observations. They rank the importance of previously learned concepts on several levels and constantly integrate previously learned concepts to acquire new, more complicated ones. DNNs have proven effective in a variety of applications, including video games, search engines, advertising, face recognition, and picture recovery.

On the other paper, they use the Mobilenet V2 from the FaceNet model which optimization of the loss function and utilized the improved model architecture. In addition to using Depthwise Separable Convolutions, MobileNet V2 suggests Inverted

Residual Blocks (shortcut links between bottlenecks) and Linear bottlenecks. Real-time and lightweight networking are becoming more and more important in the age of mobile networks. Unfortunately, due to an excessive number of parameters and computations, many identity networks are unable to meet the real-time criteria. In order to address this issue, the suggested approach that makes use of MobileNetV2 as its backbone performs better in the database of features and facial expressions than other contemporary approaches. This project will use the PYQT5 which is utility tool for designing the graphical user interface. This is an interface for Qt, one of the most well-known and significant cross-platform GUI libraries, created in Python. It also use some deep learning framework that very easy to use and powerful python libraries amd learning model such as Keras, Tendorflow, and MediaPie. The four primary processes in a face recognition process are typically face detection, face normalisation, face stamp extraction, and matchings. Multiple video streams can be handled by a Jetson Nano face recognition system. The camera module and Jetson Nano are connected so that the captured camera photos can be stored. Additionally, Jetson Nano allows smart device Internet connectivity. The feature map on which the detection from the input photos is based is called MobileNetV2. Then, to improve the performance of the back-end detection network, these feature maps with various scales are upgraded by the FPN, FPN module at the network's output. SSD detectors based on the FPN and MobileNetV2 backbone were used for facial recognition. Afterwards, ArcFace converts an image of each person's face into a vector of 512 values that indicate the key characteristics of the face. This vector is known as an embedding in machine learning. The following stage involves using a classifier to determine how far apart facial features are in order to differentiate between many identities. For the deep learning process the system need to training the dataset consist of the 60 images each

of the person for the facial authentication model. Each person that need to be register will need the images of their face from the different angle. The system will test and determine the ArcFace model against LBP and DLIB using the identical data set, four frames for each class. ArcFace has fast frame rates (FPS) and good accuracy (95–97%), respectively.

There are various benefits of using MobileNetV2 in a deep neural convolutional network for facial recognition detection. When used for tasks like face detection and recognition in security systems, MobileNetV2 is highly helpful because it lowers latency and increases processing speed. The SSD model, which is renowned for its quick and effective object identification capabilities, may be integrated with MobileNetV2 to enable the system to extract feature maps quickly and predict objects with accuracy. Furthermore, MobileNetV2 is appropriate for low-resource mobile and embedded devices due to its lightweight architecture and real-time processing capabilities, which guarantee optimal performance with less computational effort. The face recognition system performs better overall when MobileNetV2 and SSD are combined, offering a dependable and effective solution for security apps and smart home control systems.

For the paper that propose the personal assistant that using the Raspberry Pi, it proposed the voice assistant that use the speech recognition to recognize human voice and make the virtual assistant response to the user. This technology runs on a tiny computer known as a Raspberry Pi and offers a hands-free user experience. This tool assists the user with daily duties by acting as a personal assistant. This gadget is lightweight and appropriate for use at home, at business, etc. Speech recognition is the basis for the system's operation. First, the user turns on the Raspberry Pi gadget, which then welcomes the user. Upon system starting, the programme code is automatically

executed (auto-start) by means of a Python script. The system receives speech input through the microphone, records it until a pause is detected to indicate that the user has completed the request, and then converts it to text so that the computer can interpret it. The method is designed in such a way that it receives the request, looks up the keyword, and outputs the proper text. The voice-recognition module then converts the text to speech. According to the concept and algorithm it was created with, the voice assistant operates. The system greets the user at startup based on the time of day and has a power switch button. The system is given commands that are labelled with certain keywords, sometimes known as wake-up words. The system then runs the relevant Python code block after recognising these keywords. A number of modules use the API capability to get data from websites like Wikipedia, Pyowm, Oxford Dictionaries, News, IMDb, and others. The voice assistant answers in a flash, efficiently, and precisely enough. In order for the system to reply to the user, it needs a steady internet connection. It can operate without one, but it might not be fully functional because certain data is retrieved from the internet. The voice assistant apologises to the user and requests that they repeat the command if the system is unable to identify the relevant match from the keywords in the voice command. A module called SpeechRecognition makes it possible to translate text to voice and speech to text by recognising human speech. It needs audio input, and this speech recognition module can easily configure the microphone without the need for extra packages like pyaudio, which are needed in many other Python packages, or writing multiple scripts to access the microphone's input. A wrapper library for the openweathermap web API is called Pyowm. We can communicate with the following API using it. It can be used to explore satellite pictures, obtain data on air pollution, receive soil data, and obtain current and predict meteorological information.

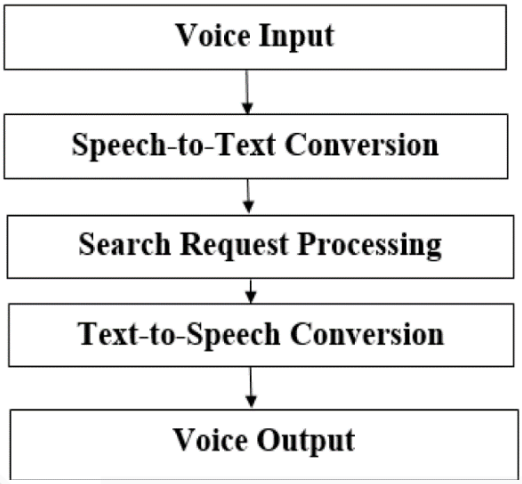

Figure 2.9: System Flow Process

Algorithm	Dimension of data	Amount of transfer data	accuracy	Transfer Learning	Performance	Database Efficiency
Principal Component Analysis on Raspberry Pi (PCA)	Low	Relative smaller	Lower	Less applicable	Faster (computationally efficient)	Lower storage requirements
Convolutional Neural Network on Raspberry Pi (CNN)	High	Very Large	Higher	Highly effective	Slower (computationally expensive)	Higher storage requirements
Local Binary Patterns (LBP)	Moderate	Moderate	Moderate	Limited applicability	Faster (efficient)	Moderate storage requirements
Deep Neural Network on Smart Home (DNN)	High	large	High	Highly effective	Varies depending on network complexity	Varies depending on network complexity
MobileNet V2 on Arduino	Lower	Large	High	Effective	Faster than larger CNN	Moderate storage requirements (compared to larger CNNs)

Table 2.2: Comparison Between Algorithms

2.3 Critical Literature Review

The goal of the research is to make the prototype of the virtual home assistant for the better performance and more secure. For the prototype of this project, I propose to use the Raspberry Pi for the prototype. The reason for the choice of Raspberry Pi is cost-effective since it is cheaper than the most personal computer. This device also more versatility because it can use for wide range of application and also Raspberry Pi has the compact size and it also can be easy to maintain.

For the algorithm that I want to use for the face recognition is Convolutional Neural Network (CNN) and Local Binary Patterns (LBP). The reason for choice CNN algorithm is it has high dimension of data and amount of transfer data and it can make the accuracy face recognition higher than other algorithms. Transfer learning for this algorithm is also high effective.

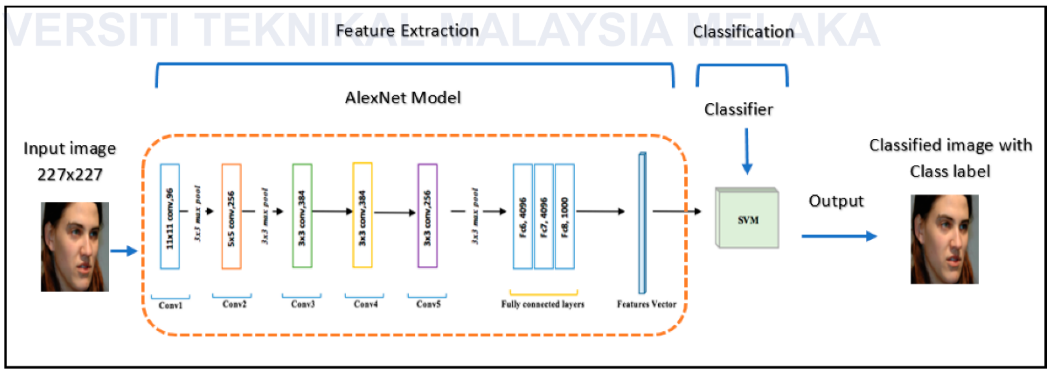

Figure 2.10: The Model of Convolutional Neural Network

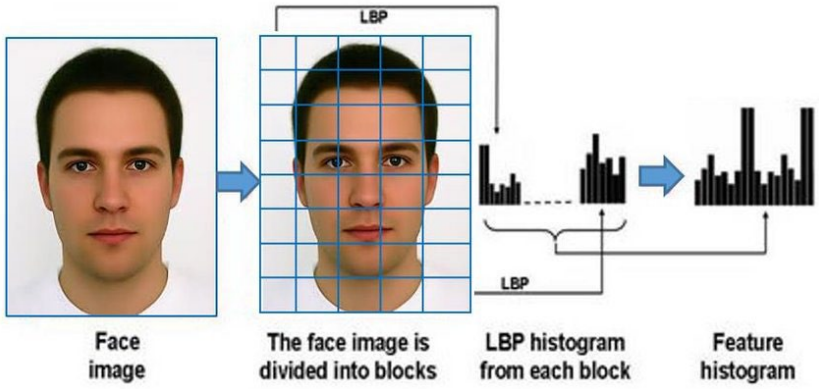

Figure 2.11: The Model of Local Binary Pattern Algorithm

But the CNN algorithm has the limitation which is the performance of this algorithm is slower than all the algorithms. So the way to improve this algorithm is by combine this algorithm with the Local Binary Pattern into this project. This can make the performance for the face recognition can be faster. The reason of the choice LBP algorithm is the performance of the algorithm is faster than other algorithms. this can be done by referring table 2.2 above.

2.4 Conclusion

Based on the table 2.2 we can conclude that this project will be referring the algorithm Convolutional Neural Network (CNN) and Local Binary Patterns (LBP). The article of this algorithm will be referring from the International Journal For Science Technology And Engineering, 30 Apr 2023.

This article uses the Convolutional Neural Network for the algorithm for the face recognition and the prototype for this research also use the Raspberry Pi. This system also will be implemented to the smart home for the security to make sure the system can be more secure. For the methodology for my project will be referring to this article.

Chapter 3

Project Methodology

3.1 Introduction

This chapter will present the methodology that will be use in this project. Methodology is the process that will be use as the guide when develop the virtual home assistant using face recognition. The approach that uses for this project is rapid application development model. rapid application development model is a process for the developing the system that involves creating a prototyping that need to test and making the modifications to get the better result for the prototype.

3.2 Methodology (RAD)

Rapid application development model is an approach to develop the system that focus on the creating project quickly and iteratively. This methodology focusses on the prototyping, and it will allow the user to test the system and it can refine the project based on the feedback from the user. This methodology is the quicker development with shorter planning stages and also allow faster adaption of the changing requirements.

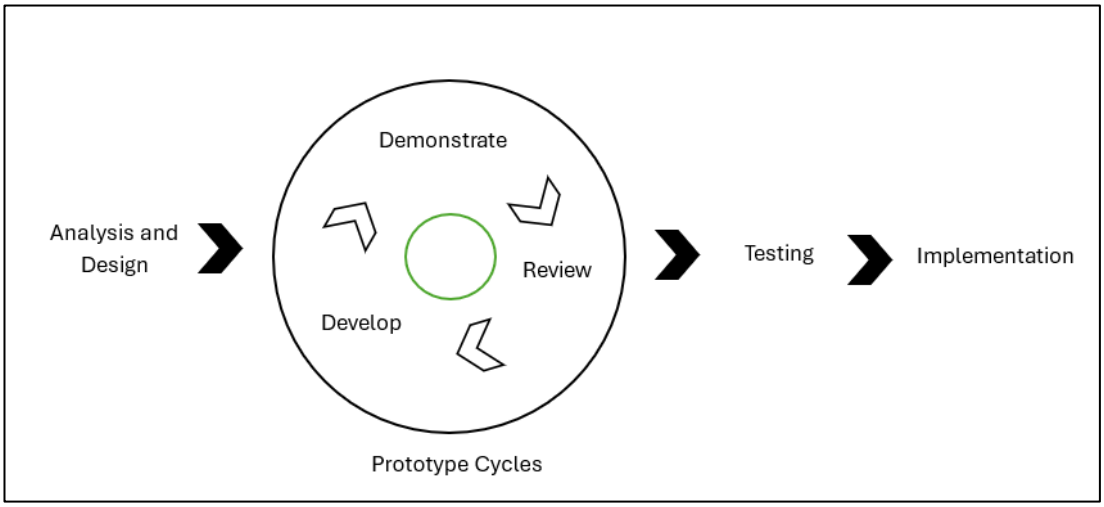

Figure 3.1: Rapid Application Development Model

3.2.1 Analysis and Design

In this process, the focus is on the analysis and gather the important information to develop the virtual home assistant using face recognition project. This includes the software and hardware that required in this project.

i. Software and algorithm requirement
    a. OpenCV
    b. Raspberry Pi OS 6.1
    c. Convolutional Neural Network (CNN)
    d. Local Binary Patterns (LBP).

ii. Hardware requirement
    a. Raspberry Pi Model 5
    b. Raspberry Pi camera
    c. External Microphone
    d. External Speaker
    e. SD card

During the design phase, all the requirement that appropriate to this prototype will be obtained. As stated in Chapter 2 above, all the design and implementation for this prototype will be referred to 'International Journal for Science Technology and Engineering', dated April 30, 2023.

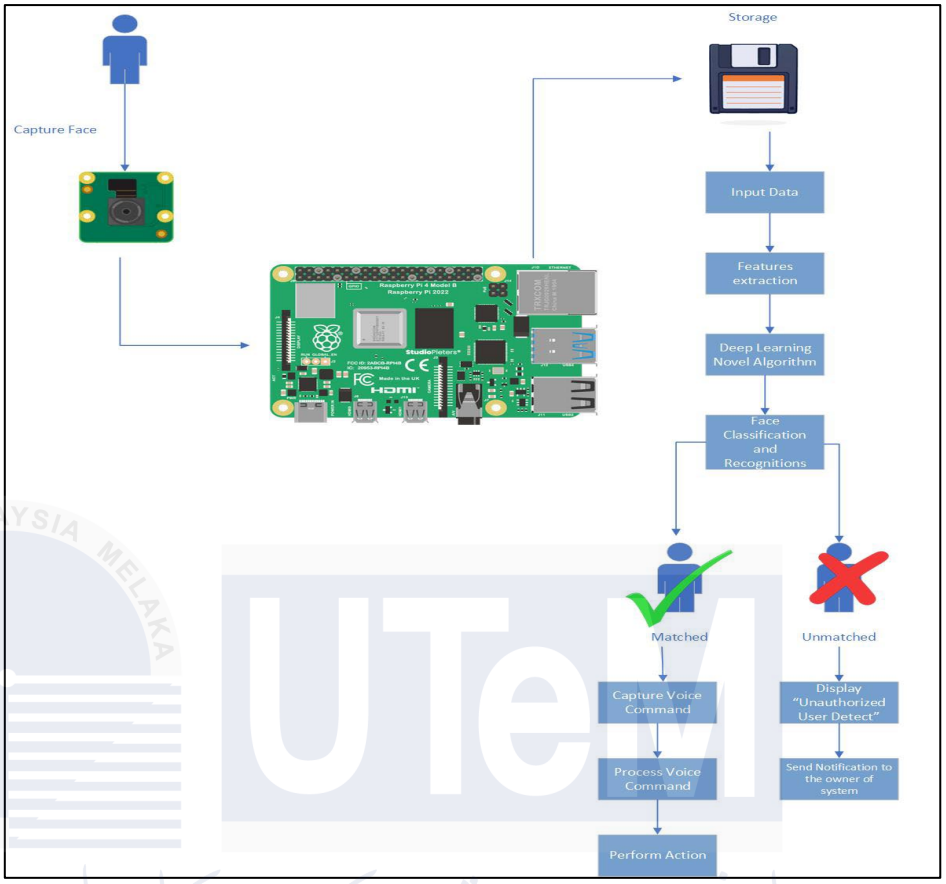

Figure 3.2: Physical Design for The Project

3.2.2 Development Process

In this process, the development of this process is assembling the necessary and installing the operating system and software that appropriate with the Raspberry Pi. For the security camera that install into the Raspberry Pi, the algorithm that configured is referring to International Journal for Science Technology and Engineering. The virtual home assistant will be set and configured to perform the specific task.

3.2.3 Demonstrate Process

In this process, the prototype that has been configured will be demonstrated to the users that will be test the program. The information and feedback on the program will be gathered from the users. This information will be use to identify the areas for improvement, add any remaining requirements, and confirm that the system meets the project requirements. This step is the crucial step because the project's development will guide by the information gained from the users.

3.2.4 Review Process

In this process, information and feedback that receive from the user will be reviewed, and the system adjustments will be made accordingly. The prototype will be evaluated to ensure it meets user requirements and fits with the project scope. The project will be reviewed from the design phase through configuration until the prototype meet the user requirements.

3.2.5 Testing Process

In this process, the completed project that has been evaluated by the users by passed the demonstration phase will be tested. The prototype that satisfies user feedback and have the minimal disadvantage s will be checked for any failures or error and the prototype will be tested before implementation with the real users. The project will be verified against the requirements to ensure all the functions meet user expectations.

3.2.6 Implementation Process

In this process, real users will be trained on the system after the testing phase. The users will operate the project deployed on the real devices, with all functions have been tested. In this phase, the system will be monitored for the error, or the bug and the maintenance will be performed as needed.

3.3 Project Milestones

In this project milestones, it will show on every task that has been establish on the develop the virtual home assistant using face recognition. The table 3.1 below show the project milestone that has been planned.

Activity	Responsibility	Date Start	Date End
PSM 1			
Requirement Analysis	**Student**	**Week 1**	**Week 2**
Design the prototype	**Student**	**Week 3**	**Week 5**
Hardware assembling	**Student**	**Week 3**	**Week 9**
Building prototype	**Student**	**Week 6**	**Week 14**
Prototype evaluation	**Student and supervisor**	**Week 8**	**Week 14**

User Evaluation	**Student, evaluator, and supervisor**	**Week 15**	**Week 15**
PSM 2			
Develop prototype based on evaluation	**Student**	**Week 1**	**Week 2**
Prototype Evaluation	**Student and supervisor**	**Week 2**	**Week 6**
Build full System	**Student**	**Week 3**	**Week 5**
Testing the system	**Student**	**Week 4**	**Week 6**
User Evaluation	**Student, evaluator, and supervisor**	**Week 7**	**Week 7**

Table 3.1 Project Milestones

Phase	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15
Analysis and Design the Prototype															
Develop the Prototype															
Demonstrate the Prototype															
Review the Prototype															
Testing System															
Implementation System															

Table 3.2 Project Milestone Gantt Chart

3.4 Conclusion

In conclusion the methodology that selected for this system is Rapid Application Development model. The decision for choose the RAD model is this approach will be priorities for the development and build the prototype. This approach can quickly make the review on the system in the middle of the development and can redevelop the system based on the user feedback without starting from the scratch. By using this approach will make the final system that develop can meet the user requirement and mad the system more user-friendly which is very important when we develop the system. The design that we decide to use is by referring the research from International Journal For Science Technology And Engineering, 30 Apr 2023.

Chapter 4

ANALYSIS AND DESIGN

4.1 Introduction

In this chapter, the project's requirements and problem will be handled in this chapter. The explanation on the necessary software and hardware will be explained more deeply in this chapter. This chapter also will cover the design approach and the flow chart of this project.

4.2 Problem Analysis

The problem that faces in the develop virtual home assistant by using face recognition is the accuracy of the face recognition system to recognise the user face. The system has succeeded in the recognizing the user's face but when tested on the unregistered users, the system sometime still can recognize the unregistered user as the owner of the system. This challenge is cause by the lack of the dataset that have collect and trained by the system. The dataset that has been collect for the trained to the system sometime does not follow the requirement to make the system more accurate. The requirement that will make the system more accurate for the face recognition is the lighting condition, the face angle and the exposure of the facial features. Additionally, the integration the face recognition with the virtual home assistant to make sure that the function of the home assistant will unlock when it recognises the registered user also the challenge that faces in this project. Addressing all the problems in creating this system is very important to make this system more user friendly.

4.3 Requirement Tools

The application that uses in this project is OpenCV and python. OpenCV is the open-source computer vision library software that common use in the face recognition that provide the image processing, video processing and perform the real time computer vision. The python is use for the configuration in this project and it can integrate with the OpenCV in the configuration. It compatible with the Raspberry Pi because it has the Python bindings. It can easily to test the face recognition by using the OpenCV and Python on the Raspberry Pi.

4.4 Environment Setup

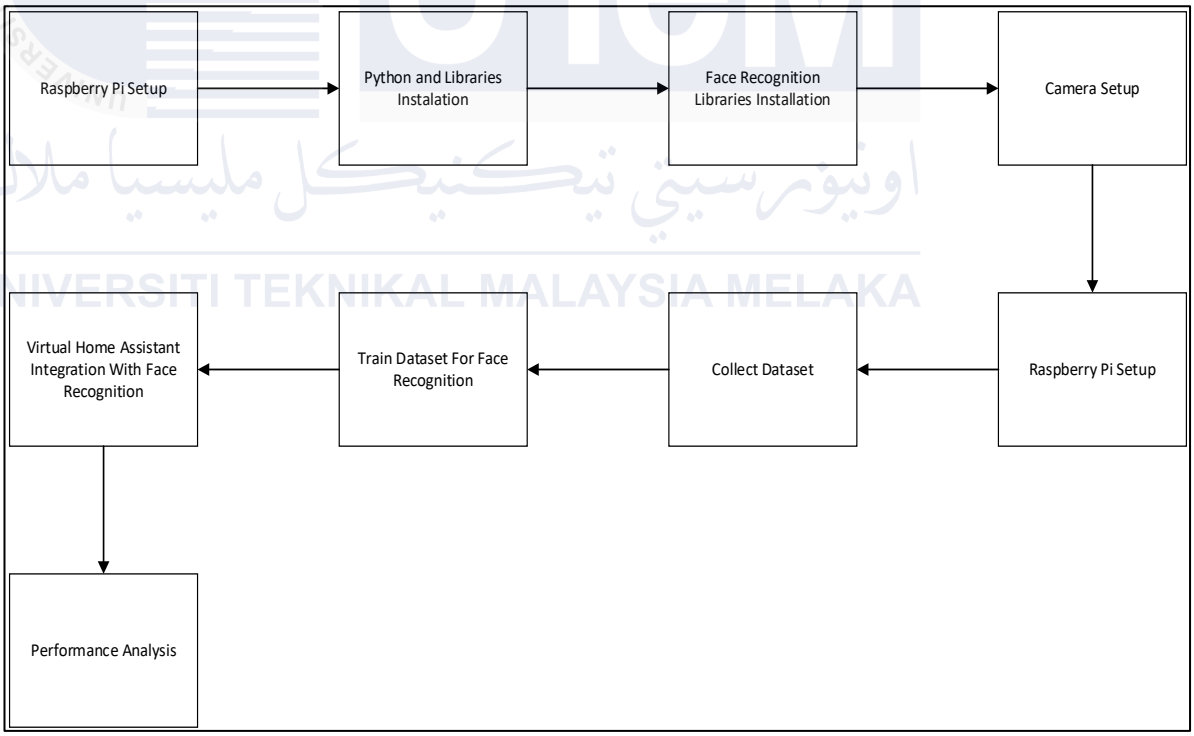

Figure 4.1 Environment Setup

4.4.1 Raspberry Pi Setup

There is several hardware that need to setup for the virtual home assistant using face recognition. It includes operating system that mount on the SD card, setup the power supply and attach the camera on the Raspberry Pi board. After mount the operating system, the SD card will be insert into the Raspberry Pi board and turn it on. After that apply the setting by setup the time zone, language and the network connection.

4.4.1.1 Hardware Design

In the hardware design, the hardware that need to be use in this prototype will select carefully according to their function and features, such as the resolution of the camera and the specification of the Raspberry Pi board. The prototype also will attach to the lcd screen where the result of the face recognition will be displayed on the screen.

4.4.1.2 Raspberry Pi 5 Model B

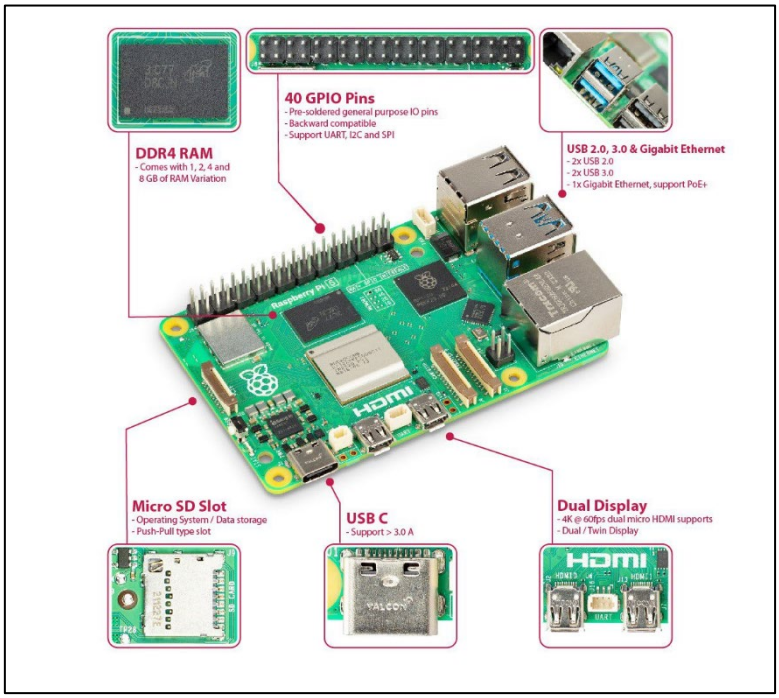

Figure 4.2 Raspberry Pi 5 Model B

The Raspberry Pi 5 is the powerful single-board computer where the device are equipped with the processor that deliver a 2-3x increase in CPU performance comparing to Raspberry Pi 4. This model comes with various RAM options from 4GB to 8GB LPDDR4X-4267 SDRAM. This make the Raspberry Pi operate smoothly and improve the performance. This board also come with Dual-band 802.11ac Wi-Fi for the wireless connection. The USB port that comes to this board supports USB 3.0 and USB 2.0. This made the data transfer more quickly with up to 10 times faster.

4.4.1.3 USB Webcam

Figure 4.3 Webcam

The webcam is a 30 FPS frame rate that use for the system to scan the user face for the face recognition. It comes with 720p where the picture that snap by using this camera become sharp and smooth. This camera easy to implement in the Raspberry Pi without using any driver, just connect to the board by using USB port.

4.4.1.4 LCD

Figure 4.4 LCD

The LCD that come with size 16×2 is a small module that can display the 16 character by 2-line display in the LCD. It can display the simple text where it can display the instruction or the result to the user.

4.4.2 Software Design

The virtual home assistant by using face recognition will use several software while develop this prototype. It use the python programming language by integrate it with OpenCV to make the accurate face recognition. The system will run by using the Raspbian OS 64 bits that mount in the SD card. The virtual home assistant will integrate with the ChatGPT API key to give the information that provide from the user. This combination of the software will give the system more user friendly and better performance for the secure virtual home assistant.

4.4.2.1 Raspbian OS

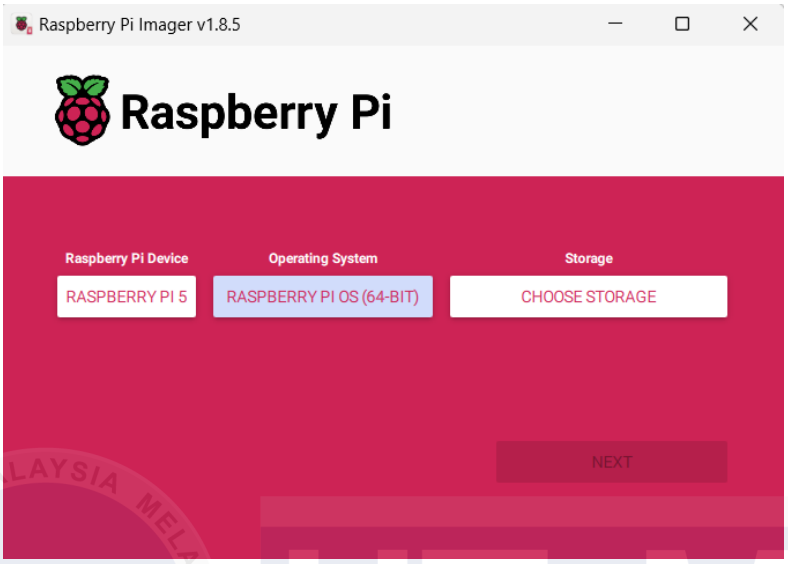

Figure 4.5 Raspbian OS

Raspbian OS is the Linux-based operating system that designed for the Raspberry Pi. It offer the user-friendly interface that can use for develop various system such as virtual home assistant. This OS is built for the compatible with the Raspberry Pi device and ensure it operate with the smooth and efficient performance to running the face recognition.

4.4.2.2 Python

Figure 4.6 Python

To develop the virtual home assistant using face recognition, the programming that suitable for the development is python. Because of it easy to access and simplicity to the face recognition, the module will be easier to implement and easy to get the

accurate result of the identifications. OpenCV libraries which use for the image processing can integrate with the python language make it more easy to implement.

4.4.2.3 OpenCV Libraries

Figure 4.7 OpenCV Libraries

OpenCV is the necessary open-source library that use for the image processing and computer processing. For this project, it will give the accurate face recognition result, give the access to the authorized user based on the facial detection. This software is compatible with the python and Raspberry Pi, it is the crucial tools that need to implement to get the reliable and secure face recognition that integrate with virtual home assistant.

4.5 System Architecture

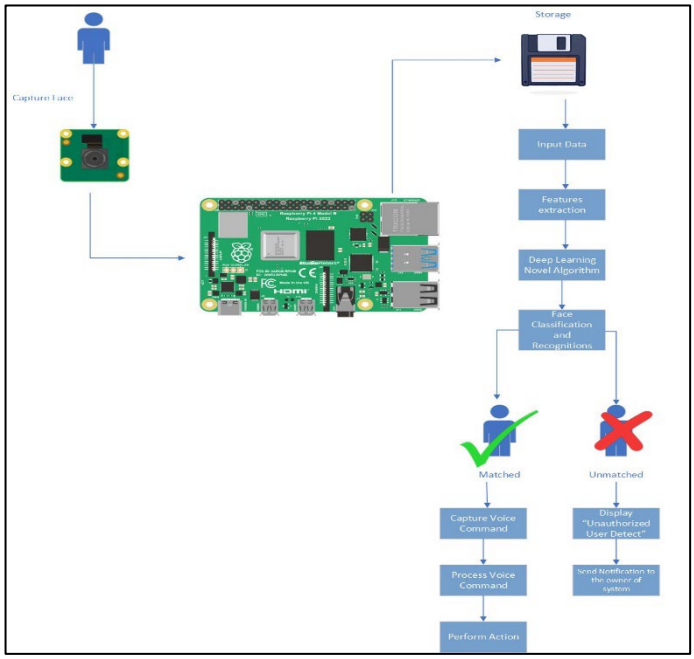

Figure 4.8 System Architecture

The system architectures are focus on the Raspberry Pi board where the main function processing is served in this board. The camera will attach to the Raspberry Pi board to capture the face for the face recognition then the board will process the face to determine is the face is recognised as the user or the unauthorized user. The 16x2 LCD screen also attach to the board to show the result of the system. The integration of the ChatGPT artificial intelligence into the virtual home assistant by using the API key to get the real-time response to the users.

4.6 Flowchart Design

The overall system flowchart will be show in the figure 4.9 where all the flow of the system will be described in that flowchart. The system first will capture the user face then the system will process the face by comparing to the dataset that has been stored and capture by the system. then if the system detects the authorized user, the system will unlock the function that can be provide for the virtual home assistant such as unlock the door. The system also can use the voice command to interact to the system and the system can perform the action. Is the system detect the unauthorized user, it will display “Unauthorized User Detect” and will send the notification to the owner of the system.

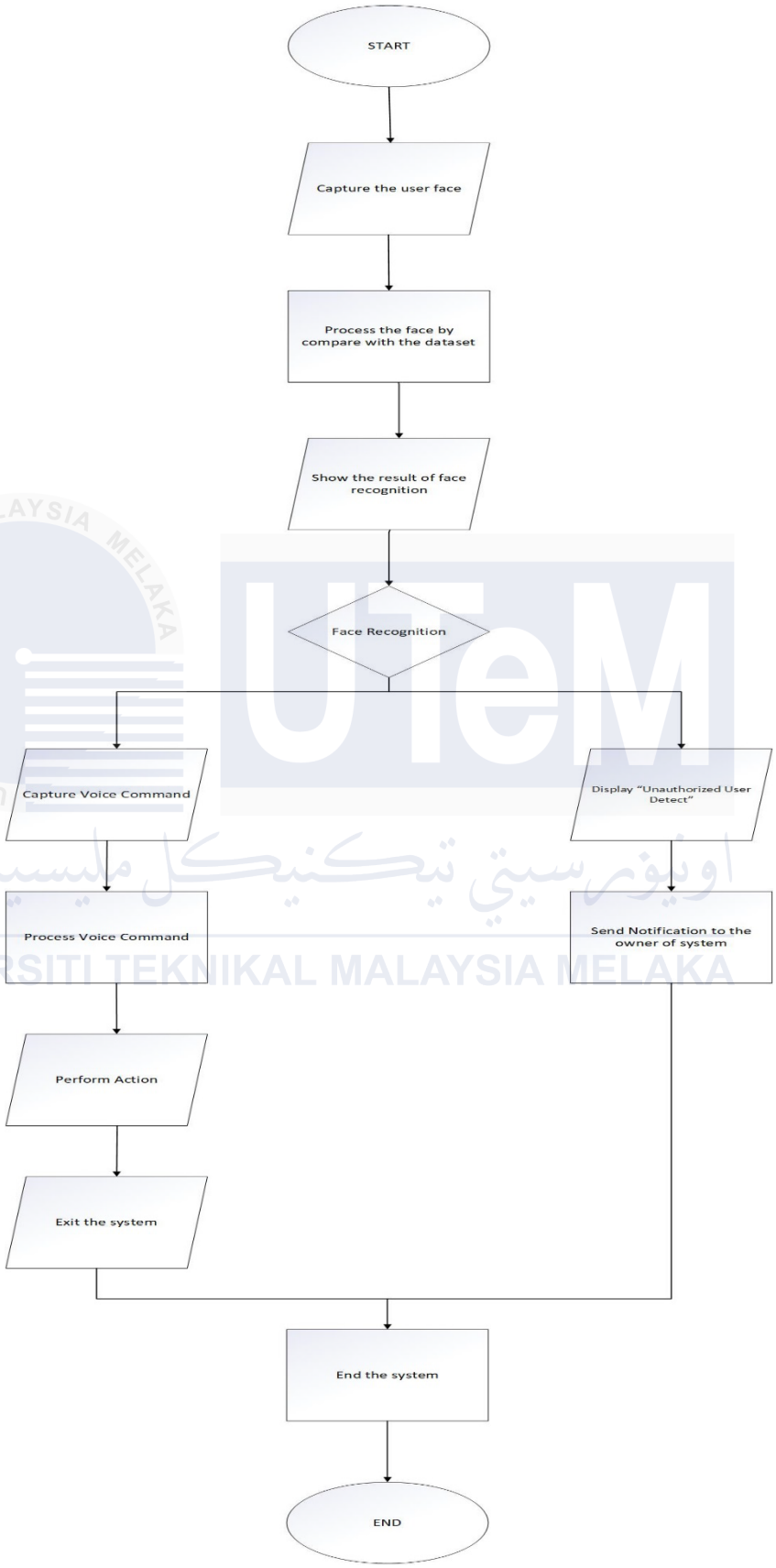

Figure 4.9 Flowchart for The System

4.7 Conclusion

For the conclusion, the virtual home assistant that integrate with face recognition can improve the security mechanism and can improve the user experience to get more better. The hardware that wants to implement in this system will be choose carefully to ensure that it can operate smoothly, and the user will get the better performance. The chosen of python that can integrate with the OpenCV and the Raspbian OS that choose for the operating system are also important for the software can work together to create the efficient solution for the system. this system will provide an efficient and modern access control because it can be adapted to different and various applications in the development process and focus on dependability and simplicity of use.

Chapter 5

Implementation

5.1 Introduction

In this chapter will focuses on the development of the virtual home assistant that can integrate with the face recognition. The virtual home assistant will be using the google generative ai based and using the API key of the Gemini Google AI. This phase will include several codes that use for developing this project.

5.2 Hardware Component

i. Raspberry Pi 5
ii. Raspberry Pi USB-C Power Supply
iii. USB Camera
iv. Portable speaker and microphone

5.2.1 Hardware Installation

The main hardware that uses for this project is Raspberry Pi 5. The USB camera will be attached to the raspberry pi for the face recognition function and the portable speaker and microphone will be connected to the raspberry pi to get the input from the user and give the output from the system.

5.3 Software Installation Setup

To make sure that the code can run without any error some of the library need to be install for this system to make sure that the code can be implemented. This includes installing OpenCV, Pi Audio library, face recognition library, speech recognition library and google generative AI library.

5.3.1 Library Installation

Figure 5.1 show that the configuration for the update raspberry pi. This is the crucial step that need to be run first to make sure that the raspberry pi operating system gets the latest packages and version of the latest security patches.

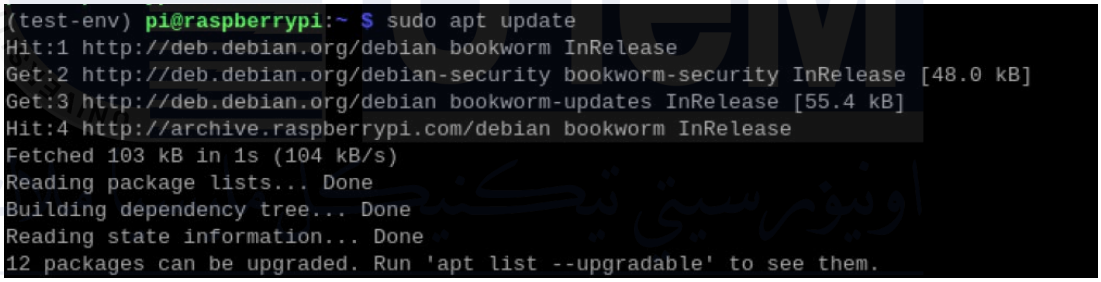

Figure 5.1: Update Configuration

The face recognition library needs to be installed by using “pip install face_recognition” when developing the face recognition system to ensure that the system can provides the high-level task for the recognition the user face. This library will include the ‘dlib’ library which this library will provide the cutting-edge for the face recognition and detection algorithm.

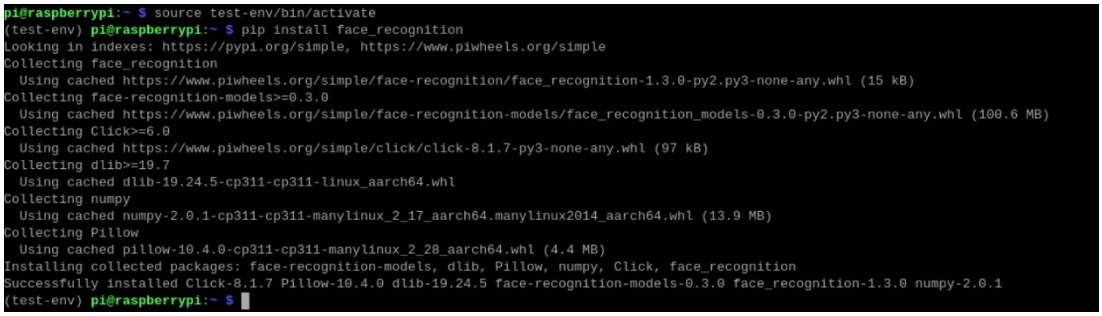

Figure 5.2: Installation of Face Recognition Library

The “imutils” library need to be install along with the face recognition library where it is the common use in the image processing, and it provides the convenient utilities for the images processing task. This library functions is for resizing, rotating, translating and displaying images.

Figure 5.3: Installation of Imutils Library

For the “opencv-python” library is the crucial library that need to be install when developing this project. This is because the OpenCV is the comprehensive library that use in the image processing and video processing.

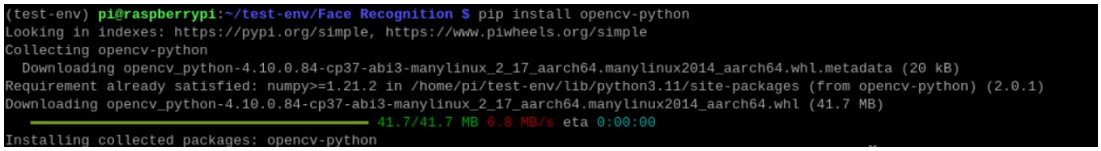

Figure 5.4: Installation of OpenCV Library

This project will use the API key from the Google Gemini AI model for the virtual home assistant get the information for the response to the user. To make sure that API key can integrate in this code, “google-generativeai” library need to be install.

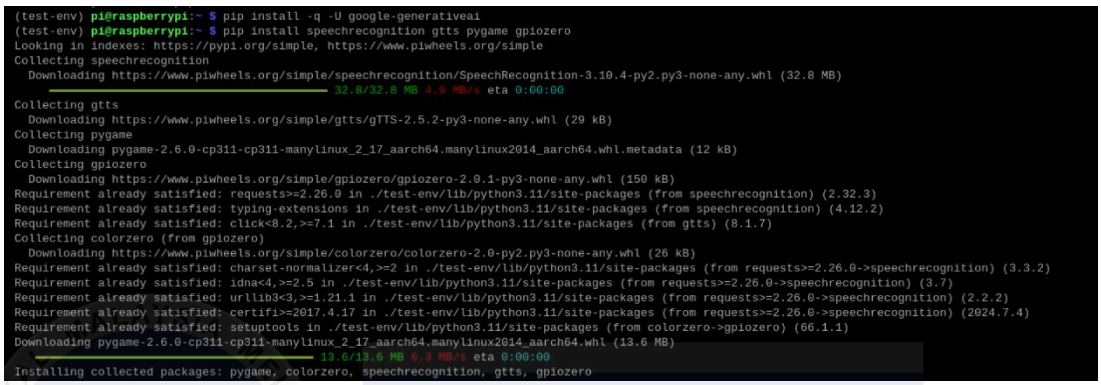

Figure 5.5: Installation of “Google-Generativeai” library

To make sure that the virtual home assistant can generate the audio and receive the speech directly from python, “portaudio19-dev” library, “pyaudio” library and “speech_recognition” library need to install in this project. The “portaudio19-dev” library is for the development of the PortAudio package and it also the dependency for the “pyaudio” library. The “pyaudio” library use for record and play the audio direct from the python and “speech_recognition” library use for the voice command that can be depends on “pyaudio” library.

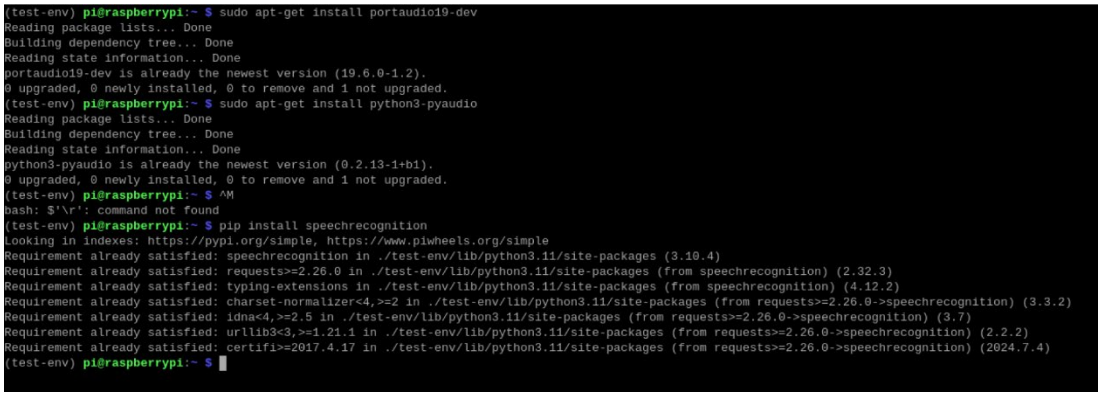

Figure 5.6: Installation of “pyaudio” and Others Library

To convert the text-to-speech that receive from the python, “pyttsx3” library will be required in this system. It will allow the program to convert the text that get from the output into the voice.

```
(test-env) pi@raspberrypi:~ $ pip install pyttsx3
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: pyttsx3 in ./test-env/lib/python3.11/site-packages (2.90)
(test-env) pi@raspberrypi:~ $ sudo apt-get install espeak
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
espeak is already the newest version (1.48.15+dfsg-3).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
(test-env) pi@raspberrypi:~ $ sudo apt-get install Flac
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package Flac
(test-env) pi@raspberrypi:~ $
```

Figure 5.7: Installation of “pyttsx3” Library

5.3.2 Camera

The next step for setting up the environment of virtual home assistant using face recognition is the configuration of the camera. The portable USB camera will be attached to the raspberry pi USB port. For the latest raspberry pi OS, it has the auto enable for the camera interface unlike the previous version, it must be enabled the camera interface. To capture the image of the user face we will use Python coding that provide the library such as the OpenCV which enable access to the camera module and frame capture for the processing. For this system the capture image coding file has been name as face_shot.py and save as the python file.

```python
import cv2

name = 'Afiff' #replace with your name

cam = cv2.VideoCapture(0)

cv2.namedWindow("press space to take a photo", cv2.WINDOW_NORMAL)
cv2.resizeWindow("press space to take a photo", 500, 300)

img_counter = 0

while True:
    ret, frame = cam.read()
    if not ret:
        print("failed to grab frame")
        break
    cv2.imshow("press space to take a photo", frame)

    k = cv2.waitKey(1)
    if k%256 == 27:
        # ESC pressed
        print("Escape hit, closing...")
        break
    elif k%256 == 32:
        # SPACE pressed
        img_name = "dataset/"+ name +"/image_{}.jpg".format(img_counter)
        cv2.imwrite(img_name, frame)
        print("{} written!".format(img_name))
```

Figure 5.8: Coding For Capture Image

```python
    elif k%256 == 32:
        # SPACE pressed
        img_name = "dataset/"+ name +"/image_{}.jpg".format(img_counter)
        cv2.imwrite(img_name, frame)
        print("{} written!".format(img_name))
        img_counter += 1

cam.release()

cv2.destroyAllWindows()
```


Figure 5.9: Coding For Capture Image

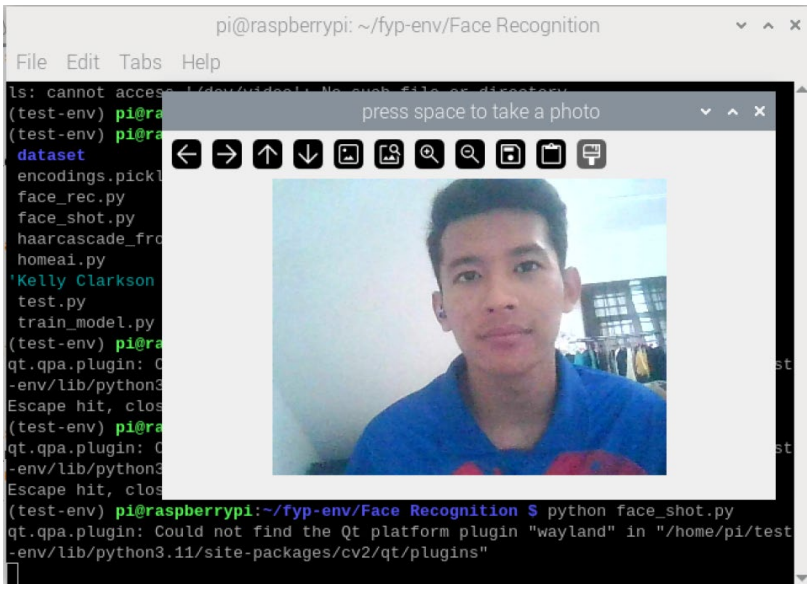

Figure 5.10: Camera Ready to Capture Image

Once the USB camera is set up to the raspberry pi and make sure that the camera can operate without any problem, the face_shot.py file need to be run to capture the user face for the dataset for the face recognition. The dataset that has been capture will be stored in the dataset folder and name it as the username.

5.3.3 Face Recognition Dataset

The face recognition dataset is the important in this project for the training and the recognize the user faces. The dataset is storing all the user interface that has been captured. All the user face will be stored by their name to make sure that the system will recognize the user accurately.

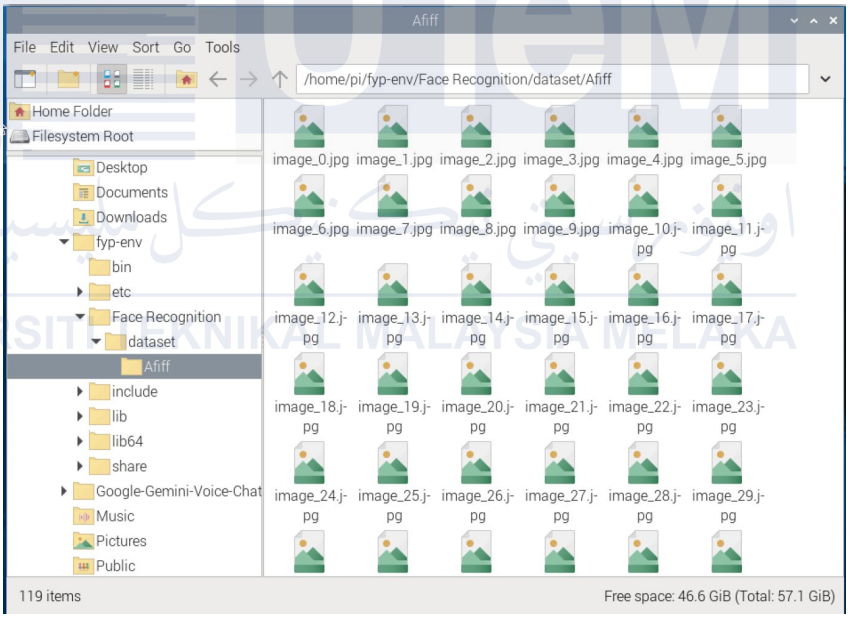

Figure 5.11: User Face Stored in Dataset

At the code, the location for the image to be save after capture the user face will be declared by giving the specific path of the folder as show on the figure 5.12.

```
elif k%256 == 32:
    # SPACE pressed
    img_name = "dataset/"+ name +"/image_{}.jpg".format(img_counter)
    cv2.imwrite(img_name, frame)
    print("{} written!".format(img_name))
    img_counter += 1
```

Figure 5.12: The Path of Saved Images

The name for the user's folder that saved the face images is declared at the first line of the code as shown in the figure 5.13.

```
1  import cv2
2
3  name = 'Afiff' #replace with your name
4
5  cam = cv2.VideoCapture(0)
```

Figure 5.13: User's Folder for the Saved Images

5.3.4 Training Dataset

The next step in setting up the environment is training the dataset that has been capture in the dataset folder. The capture images that saved in the dataset is used to teach the algorithm on how to recognize the user that has been register to the system based on their face that has been captured.

```
1  #! /usr/bin/python
2
3  # import the necessary packages
4  from imutils import paths
5  import face_recognition
6  #import argparse
7  import pickle
8  import cv2
9  import os
10
11 # our images are located in the dataset folder
12 print("[INFO] start processing faces...")
13 imagePaths = list(paths.list_images("dataset"))
14
15 # initialize the list of known encodings and known names
16 knownEncodings = []
17 knownNames = []
18
19 # loop over the image paths
20 for (i, imagePath) in enumerate(imagePaths):
21     # extract the person name from the image path
22     print("[INFO] processing image {}/{}".format(i + 1,
23         len(imagePaths)))
24     name = imagePath.split(os.path.sep)[-2]
25
```

Figure 5.14: Code for Training Dataset

```
    # load the input image and convert it from RGB (OpenCV ordering)
    # to dlib ordering (RGB)
    image = cv2.imread(imagePath)
    rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    # detect the (x, y)-coordinates of the bounding boxes
    # corresponding to each face in the input image
    boxes = face_recognition.face_locations(rgb,
        model="hog")

    # compute the facial embedding for the face
    encodings = face_recognition.face_encodings(rgb, boxes)

    # loop over the encodings
    for encoding in encodings:
        # add each encoding + name to our set of known names and
        # encodings
        knownEncodings.append(encoding)
        knownNames.append(name)

# dump the facial encodings + names to disk
print("[INFO] serializing encodings...")
data = {"encodings": knownEncodings, "names": knownNames}
f = open("encodings.pickle", "wb")
f.write(pickle.dumps(data))
f.close()
```

Figure 5.15: Code for Training Dataset

Figure 5.14 and Figure 5.15 show that the code for the training the dataset to make sure that the face recognition system can recognise the user face more accurate. First in the code will import the necessary packages that need for this code to run. Then the code will list all the images path that has been stored in the dataset directory. Then it will store the facial encoding for each face detect and list all the stored corresponding name associated with each encoding. Then the system will use the OpenCV to convert the images from BGR colour space “to the RGB colour spaces that use by “face_recognition” library. Then the facial encoding will determine for each of the faces detected in the images that has been capture. Then the encoding will loop for each facial encoding that found in the images. After that the system will stores the encoding with the corresponding name and name it into the pickle files for the further face recognition task.

5.3.5 Virtual Home Assistant

The main purpose of this project is to make the virtual home assistant to be more secure by using the face recognition. The next process in the development this project is to develop the virtual home assistant and make sure that it can communicate and response to the user by using the voice command. This project will use the API key from the Google Gemini AI to connect to the google server and get the information for the response to the user question. The virtual home assistant that develops also can perform some of the function of the smart home such as lock and unlock the door and can play the song for the user.

```
import streamlit as st
import os
import google.generativeai as genai
import speech_recognition as sr
import pyttsx3
from datetime import datetime

# Initialize the text-to-speech engine
engine = pyttsx3.init()

# Set the speech rate to a slower value for clarity
engine.setProperty('rate', 140)  # Default is around 150-200, lower values are slower

# Set the volume to maximum for better clarity
engine.setProperty('volume', 1.0)

# List available voices and choose a clear English one
voices = engine.getProperty('voices')
for voice in voices:
    if 'en' in voice.languages:  # Ensure English voice is selected
        engine.setProperty('voice', voice.id)
        break

```

Figure 5.16: Code For the Text-To-Speech Engine

Figure 5.16 above show that the first part of the code for the develop the virtual home assistant. The first step is to import the required library that need for this system. To convert the text to speech to make sure that the assistant can give the voice, the system will be required the pyttsx3 library. Then the code can be modifying the voice of the virtual assistant according to the speed of the voice at the “engine.setProperty” part. The default speed that usually use is 150 to 200 but in this project will use 140 for the better speed to make sure that the output can be heard clearly. Then the next step is to

adjust the volume of the system and select the language of this system for the clear English.

```
# Test output
engine.say("Hello, this is FIFA, stands for... Facial-Recognition Friendly Intelligent Flexible Assistant.")
engine.runAndWait()

# Set your Google Gemini API key
os.environ['GOOGLE_API_KEY'] = "AIzaSyBRQMsjJzrnUzyB2ZDUZ3nZy3Gkpk26NS8"
genai.configure(api_key=os.environ['GOOGLE_API_KEY'])

# Select the model
model = genai.GenerativeModel('gemini-pro')

# Initialize chat history
if "messages" not in st.session_state:
    st.session_state.messages = [
        {
            "role": "assistant",
            "content": "How can I assist you today?"
        }
    ]

# Display chat messages from history on app rerun
for message in st.session_state.messages:
    with st.chat_message(message["role"]):
        st.markdown(message["content"])

```

Figure 5.17: Code for configuring Google API and Chat History

The next step in this code as show on the figure 5.17 is the setup of the API key to make sure that the information that provide for the virtual assistant is from the google server and the system will use the google environment to provide the information to the user. Then the code will select the “gemini-pro” model for the generating the response form the queries that get from the user. The at the “st.session_state” it will check the chat history in the stateful information and if it does not have the chat history, the system will initializes with the default assistant message.

```python
# Function to get the current date
def get_current_date():
    today = datetime.today().strftime('%Y-%m-%d')
    return today

# Function to process query and generate response
def llm_function(query):
    # Check if the query asks for the date
    if "date" in query.lower():
        response_text = f"Today's date is {get_current_date()}."
    elif "party" in query.lower():
        response_text = "Playing your favorite song."
        play_song("/home/pi/fyp-env/Face Recognition/Kelly Clarkson - Because Of You (Lyrics).mp3")
    elif "stop" in query.lower():
        response_text = "Stopping the song."
        stop_song()
    else:
        response = model.generate_content(query)
        response_text = response.text.strip()  # Clean the text before speaking it

    # Displaying the Assistant Message
    with st.chat_message("assistant"):
        st.markdown(response_text)

```

Figure 5.18: Code To Get the Date and Play Song

For the figure 5.18 show that the code for the virtual assistant to perform the function of the virtual home assistant. Gort he part “get_current_date”, it will make the system to get the real time date when the user asks for the date. The “llm_function” is the part where all the function that assign to the virtual home assistant will be implement in the code.


```python
# Function to capture voice input with feedback
def listen_to_voice():
    recognizer = sr.Recognizer()

    # Text feedback in Streamlit
    st.write("Listening for your query...")

    # Audio feedback using pyttsx3
    engine.say("What can I help you with?")
    engine.runAndWait()

    with sr.Microphone() as source:
        recognizer.adjust_for_ambient_noise(source, duration=1)  # Adjust for ambient noise
        try:
            audio = recognizer.listen(source, timeout=5, phrase_time_limit=5)
            query = recognizer.recognize_google(audio)
            st.write(f"You said: {query}")
            return query.strip()  # Clean the user's spoken input
        except sr.UnknownValueError:
            st.write("Sorry, I didn't catch that.")
            engine.say("Sorry, I didn't catch that.")
            engine.runAndWait()
        except sr.RequestError:
            st.write("There was an issue with the speech recognition service.")
            engine.say("There was an issue with the speech recognition service.")
            engine.runAndWait()
```

Figure 5.19: Code for Capturing Voice Input

Figure 5.19 above show the code for the system to capture all the voice input that get from the user. First the system will be listening for the user command where the output that system will give is “What can I help you with”. Then the system will open the microphone to capture the user’s voice. Then google speech recognition will be use for the conversion from text to audio. If the system have some problem from the hear the voice or have the unexpected error it will give the voice output for the error handling.

5.3.6 Virtual Home Assistant Integrate with Face Recognition

In this part of the code will provide the full code to implement the virtual home assistant that integrates with the face recognition. When the system detect the user the user that registered th the system, it will unlock the virtual home assistant and if system detect the unauthorized user it will lock the system and the system say “system denied system locked”.


```python
import streamlit as st
import os
import google.generativeai as genai
import speech_recognition as sr
import pyttsx3
from datetime import datetime
from imutils.video import VideoStream, FPS
import face_recognition
import imutils
import pickle
import time
import cv2
import pygame

# Initialize the text-to-speech engine
engine = pyttsx3.init()

# Set the speech rate and volume for clarity
engine.setProperty('rate', 140)
engine.setProperty('volume', 1.0)

# List available voices and choose a clear English one
voices = engine.getProperty('voices')
for voice in voices:
    if 'en' in voice.languages:
        engine.setProperty('voice', voice.id)
        break

```

Figure 5.20: Code For the Text-To-Speech Engine

Figure above show that the first part of the code for the develop the virtual home assistant. The first step is to import the required library that need for this system. To convert the text to speech to make sure that the assistant can give the voice, the system will be required the pyttsx3 library. Then the code can be modifying the voice of the virtual assistant according to the speed of the voice at the “engine.setProperty” part. The default speed that usually use is 150 to 200 but in this project will use 140 for the better speed to make sure that the output can be heard clearly. Then the next step is to adjust the volume of the system and select the language of this system for the clear English.

```
# Set your Google Gemini API key
os.environ['GOOGLE_API_KEY'] = "AIzaSyBRQMsjJzrnUzyB2ZDUZ3nZy3Gkpk26NS8"
genai.configure(api_key=os.environ['GOOGLE_API_KEY'])

# Initialize the model globally
model = genai.GenerativeModel('gemini-pro')

# Load face recognition data
encodingsP = "/home/pi/fyp-env/Face Recognition/encodings.pickle"
cascade = "/home/pi/fyp-env/Face Recognition/haarcascade_frontalface_default.xml"

print("[INFO] loading encodings + face detector...")
data = pickle.loads(open(encodingsP, "rb").read())
detector = cv2.CascadeClassifier(cascade)

# Initialize the video stream and allow the camera sensor to warm up
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()
time.sleep(2.0)

# Check if the video stream was initialized properly
if vs.stream is None or not vs.stream.isOpened():
    print("[ERROR] Could not open video stream.")
    engine.say("Error: Could not access the camera.")
    engine.runAndWait()
    st.write("Error: Could not access the camera.")
    # Exit or return from the function if needed
```

Figure 5.21: Gemini API Key Configuration and Video Stream Initialization

The next step in this code as show on the figure 5.17 is the setup of the API key to make sure that the information that provide for the virtual assistant is from the google server and the system will use the google environment to provide the information to the user. Then the system will load the facial encoding and initialized the face detector

by using the Haar cascade classifier. Then the video stream will start and it will checked for the successful.

```
else:
    # Start the FPS counter
    fps = FPS().start()

# Initialize chat history
if "messages" not in st.session_state:
    st.session_state.messages = [
        {
            "role": "assistant",
            "content": "How can I assist you today?"
        }
    ]

# Initialize pygame mixer
pygame.mixer.init()

def play_song(song_path):
    pygame.mixer.music.load(song_path)
    pygame.mixer.music.play()

def stop_song():
    pygame.mixer.music.stop()

```

Figure 5.22: Initialization For Music

The at the “st.session_state” it will check the chat history in the stateful information and if it does not have the chat history, the system will initializes with the default assistant message. The pygame library will be use in this step for the music playback and for the function start and stop the music.


```
def recognize_user():
    currentname = "unknown"
    authorized_user_detected = False
    while True:
        # Grab the frame from the threaded video stream and resize it
        frame = vs.read()
        frame = imutils.resize(frame, width=500)

        # Convert the input frame from (1) BGR to grayscale (for face detection) and (2) from BGR to RGB (for face recognition)
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

        # Detect faces in the grayscale frame
        rects = detector.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30), flags=cv2.CASCADE_SCALE_IMAGE)

        # OpenCV returns bounding box coordinates in (x, y, w, h) order but we need them in (top, right, bottom, left) order
        boxes = [(y, x + w, y + h, x) for (x, y, w, h) in rects]

        # Compute the facial embeddings for each face bounding box
        encodings = face_recognition.face_encodings(rgb, boxes)
        names = []

        # Loop over the facial embeddings
        for encoding in encodings:
            matches = face_recognition.compare_faces(data["encodings"], encoding)
            name = "Unknown"  # if face is not recognized, then print Unknown

```

Figure 5.23: Code for The Setup Frame

```
105
106             if True in matches:
107                 matchedIdxs = [i for (i, b) in enumerate(matches) if b]
108                 counts = {}
109
110                 for i in matchedIdxs:
111                     name = data["names"][i]
112                     counts[name] = counts.get(name, 0) + 1
113
114                 name = max(counts, key=counts.get)
115                 if currentname != name:
116                     currentname = name
117                     print(f"Recognized: {currentname}")
118                     authorized_user_detected = True
119                 else:
120                     authorized_user_detected = False
121
122             names.append(name)
```

Figure 5.24: Code for Recognized the User

```
124         # Display the detected face and name on the frame
125         for ((top, right, bottom, left), name) in zip(boxes, names):
126             cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)
127             y = top - 15 if top - 15 > 15 else top + 15
128             cv2.putText(frame, name, (left, y), cv2.FONT_HERSHEY_SIMPLEX, .8, (255, 0, 0), 2)
129
130         cv2.imshow("Facial Recognition is Running", frame)
131         key = cv2.waitKey(1) & 0xFF
132
133         # Exit if 'q' is pressed
134         if key == ord("q"):
135             break
136
137         # Check if a known user is recognized
138         if authorized_user_detected:
139             engine.say(f"Welcome {currentname}. Virtual Assistant unlocked.")
140             engine.say("Hello, this is FIFA, stands for... Facial-Recognition Friendly Intelligent Flexible Assistant.")
141             print(f"Welcome {currentname}. Virtual Assistant unlocked.")
142             print("Hello, this is FFIFA, stands for... Facial-Recognition Friendly Intelligent Flexible Assistant.")
143             engine.runAndWait()
144             return True  # Exit loop if a registered user is detected
145
```

Figure 5.25: Code For Output When System Unlock


```
146         # Block access if the user is unknown
147         if not authorized_user_detected and currentname == "Unknown":
148             engine.say("Access denied. Unrecognized user.")
149             engine.runAndWait()
150             print("Access denied. Unrecognized user.")
151             return False  # Block access
152
153
154     # Cleanup
155     fps.stop()
156     cv2.destroyAllWindows()
157     vs.stop()
158     return False
159
```

Figure 5.26: Code for Unrecognized User

Figure above is the code for the system to recognize the user by referring the dataset that has been trained for this system. the system will capture the frames from the video stream then it will try to recognize the user's faces. If it recognises the authorized user

it will unlock the system and if the system detect the unauthorized user it will lock the user from access the virtual assistant.

```
159
160 def llm_function(query):
161     print(f"User query: {query}") # print user input
162
163     if "date" in query.lower():
164         response_text = f"Today's date is {datetime.today().strftime('%Y-%m-%d')}."
165     elif "party" in query.lower():
166         response_text = "Playing your favorite song."
167         play_song("/home/pi/fyp-env/Face Recognition/Kelly Clarkson - Because Of You (Lyrics).mp3")
168     elif "stop" in query.lower():
169         response_text = "Stopping the song."
170         stop_song()
171     elif "unlock" in query.lower():
172         response_text = "door is unlock now"
173     elif "lock" in query.lower():
174         response_text = "door is lock now"
175     else:
176         response = model.generate_content(query)
177         response_text = response.text.strip()
178
179     print(f"Assistant response: {response_text}") #print assistant's response
180
181     with st.chat_message("assistant"):
182         st.markdown(response_text)
183
184     engine.say(response_text)
185     engine.runAndWait()
186
```

Figure 5.27: Code To Implement the Function of Virtual Assistant

For the figure above show that the code for the virtual assistant to perform the function of the virtual home assistant. At the part “get_current_date”, it will make the system to get the real time date when the user asks for the date. This system also can play the song for the user and also can lock and unlock the door for the user. The “llm_function” is the part where all the function that assign to the virtual home assistant will be implement in the code.

```
187     st.session_state.messages.append({"role": "user", "content": query})
188     st.session_state.messages.append({"role": "assistant", "content": response_text})
189
190 def listen_to_voice():
191     recognizer = sr.Recognizer()
192     st.write("Listening for your query...")
193     engine.say("What can I help you with?")
194     engine.runAndWait()
195
196     with sr.Microphone() as source:
197         recognizer.adjust_for_ambient_noise(source, duration=1)
198         try:
199             audio = recognizer.listen(source, timeout=5, phrase_time_limit=5)
200             query = recognizer.recognize_google(audio)
201             print(f"Voice input: {query}")  # Print the recognized speech
202             st.write(f"You said: {query}")
203             return query.strip()
204         except sr.UnknownValueError:
205             response = "Sorry, I didn't catch that."
206             print(f"Engine response: {response}")  # Print engine response
207             st.write(response)
208             engine.say(response)
209             engine.runAndWait()
210         except sr.RequestError:
211             response = "There was an issue with the speech recognition service."
212             print(f"Engine response: {response}")  # Print engine response
213             st.write(response)
214             engine.say(response)
```

Figure 5.28: Code For the Capture and Process Voice

```
215             engine.runAndWait()
216         except Exception as e:
217             response = f"An unexpected error occurred: {e}"
218             print(f"Engine response: {response}")  # Print engine response
219             st.write(response)
220             engine.say(response)
221             engine.runAndWait()
222     return None
223
```

Figure 5.29: Continuous Code for The Capture and Process Voice

For this part of the code the system will capture the voice that get from the user, and it will process the voice for the system to give the output to the user. If the user's speech is recognized, the system will convert it to the text and send it to perform he action for this system.

```
224 def run_virtual_assistant():
225     while True:
226         query = st.chat_input("What's up?")
227 
228         if not query:
229             query = listen_to_voice()
230 
231         if query:
232             print(f"User input: {query}")  # Print the user's query
233 
234         if query and "bye" in query.lower():
235             response = "Goodbye! Have a great day!"
236             print(f"Engine response: {response}")  # Print engine response
237             engine.say(response)
238             engine.runAndWait()
239             st.write(response)
240             break
241 
242         if query:
243             with st.chat_message("user"):
244                 st.markdown(query)
245             llm_function(query)
246
```

Figure 5.30: Code for Running the Virtual Assistant

For this part of the code it will define the main loop to run the virtual assistant until the user say the keyword to end the process. In this project when the user say keyword "bye" the system will end the process for the listening for the user input.

```
246
247 # Start the virtual assistant if a registered user is recognized
248 if recognize_user():
249     print("Authorized user recognized. Starting virtual assistant.")
250     run_virtual_assistant()
251 else:
252     response = "Access denied. System locked."
253     print(f"Engine response: {response}")  # Print engine response
254     engine.say(response)
255     engine.runAndWait()
256     st.write(response)
257
```

Figure 5.31: Code For the System to Start for Recognized User

This code will provide the system to start the virtual home assistant when it detects the recognized user and if it detects the unrecognized user, it will lock the user and end the process.

5.4 Implementation Status

The implementation for the Virtual Home Assistant using Face Recognition is moving along successfully according to the Gantt chart that was planned in Chapter 3. The development for the virtual home assistant is carried out simultaneously with the creation of the project report, covering Chapter 1 to Chapter 5.

For the completion of the project report covering each chapter is being completed and is on track to be finalized at the end of August. The development of this system is being diligent in preparing and dealing with all the errors obtained while developing this system.

For the virtual home assistant has been develop well with the several key components has already completed. The text to speech has been successfully setup and has been test for the functionality. For the face recognition has been setup and trained all the dataset. It also has been test for the user recognition where it detect the user face that has been register with the system. For the face recognition system it just waiting to be implement and integrated with the virtual home assistant system.

In conclusion, the implementation of virtual home assistant using face recognition has been done well according to the planning. The development of this system has also been done together with the final report that has been made according to the chapter that has been determined up to chapter 7. I have put in full effort to complete this project according to the timeline that has been planned and I optimized that this project can be completed according to everything that has been planned.

5.5 Conclusion

In conclusion, the implementation chapter for the Virtual Home Assistant Using Face Recognition has covered several critical aspects along of the development of this system. This chapter covered the hardware selection, software installation and development, the face recognition integration and the development of the virtual home assistant.

The hardware selection phase has been done very carefully by ensuring that all functions for each hardware can provide benefits and functions suitable for this project. The hardware that has been select are all compatible for face recognition function and for the virtual assistant.

For the software installation and development phase, the selected library that need for this project and the API key has been selected based on the research on the purpose of the functionality and the compatible on the java and the raspberry pi. The integration of virtual home assistant and face recognition has enhanced the security of the virtual home assistant by enabling the function of the virtual home assistant only for the authorized user by using their face.

The execution of hardware selection, software installation and development and the integration of the virtual home assistant with the face recognition has offered the strong foundation for the advancement for this project. With the strong foundation, it is possible to demonstrate a persistent commitment in the develop the Virtual Home Assistant Using Face Recognition.

Chapter 6

Testing

6.1 Introduction

In this chapter the discussion of the project's prototype is on how the project is been tested. This step is the crucial step in this project because testing phase is the part that to make sure that the system can manage to recognize the user and give the right permission to unlock the system. In this chapter, the main purpose of this chapter is to test the system to make sure that the system can achieve the project objective that has been determined for this project.

6.2 Testing User Authentication Using Face Recognition

The main security feature in this project is the user authentication by using the face recognition. When the user that has been registered and the dataset of the user has been add and trained, the system would be test for the functionality of the user authentication. The expected outcome for this part of the testing is to make sure that the system would be recognize the authorized user and unlock the virtual home assistant.

Test	Test Objective	Test Setup	Expected Result
Authorized User	To test the user authentication for the authorized user	The dataset of the authorized user that collected and stored in one folder and the dataset was trained. Then use the camera to test the face recognition system.	The system will recognize the user and tell the name of the user. Then the system will unlock the virtual home assistant, and the user can use the function of the virtual home assistant
Unauthorized User	To test the user authentication for the unauthorized user	Use the camera and point the camera to the unauthorized user	The system will recognize the unauthorized user as the unknow user, and it will stop the system and give the instruction to tell that the system is unlock.

Table 6.1: Testing the User Authentication

When the system is started, the system began the video stream and the interface opened a window of the camera to detect the user if the user is the authorized user or the unknow user. If it detects the authorized user the system will welcome the user and the function of the virtual home assistant were unlock.

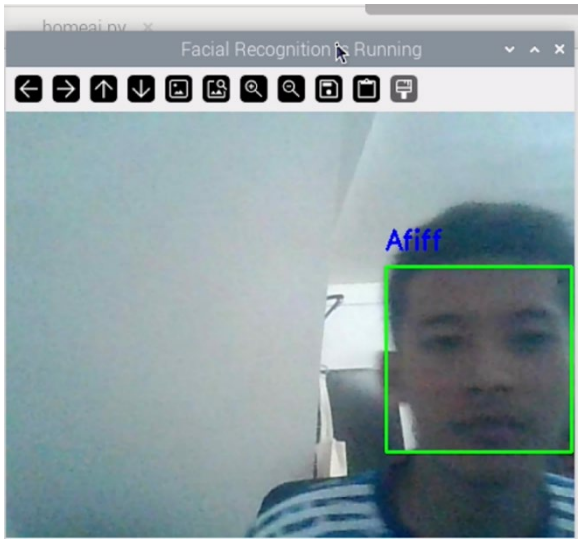

Figure 6.1: Detect The Authorized User

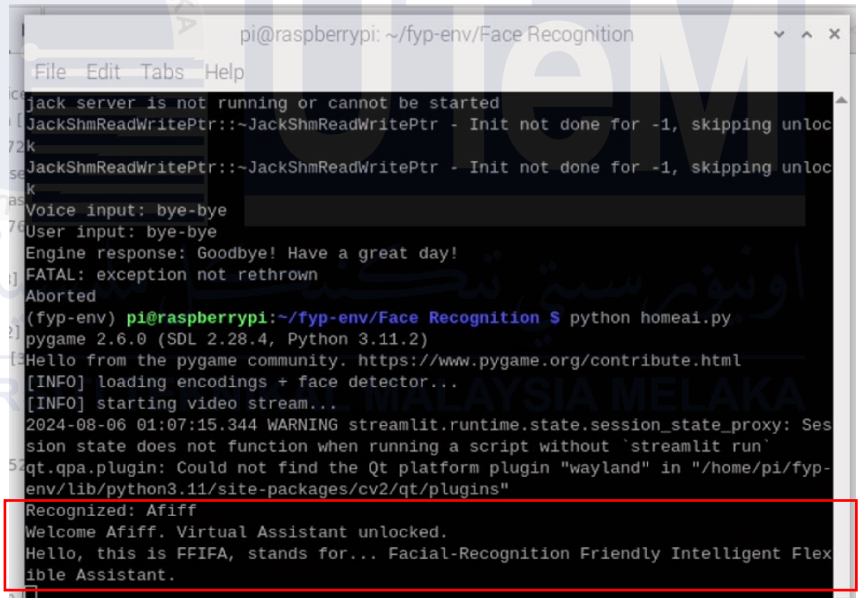

Figure 6.2: The System Welcome the Authorized User

If the system detects the unauthorized user in the camera display window, the system show that the user is unknow and the system would say “Access denied, system lock”. Figure 6.3 and figure 6.4 below show that how the system detect the unknow user and the system say the lock system function.

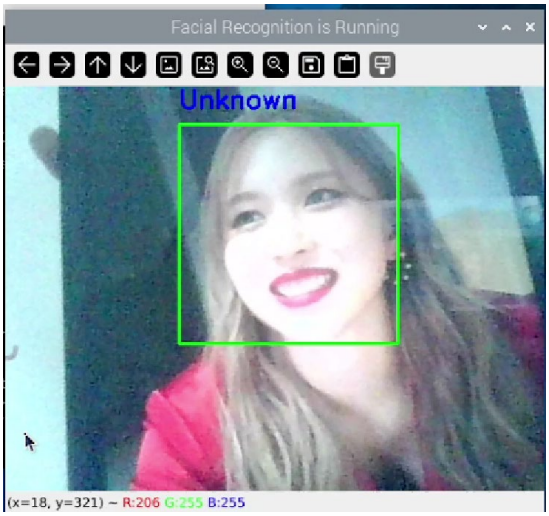

Figure 6.3: Unknow User Detect by the System

```
(fyp-env) pi@raspberrypi:~/fyp-env/Face Recognition $ python homeai.py
pygame 2.6.0 (SDL 2.28.4, Python 3.11.2)
Hello from the pygame community. https://www.pygame.org/contribute.html
[INFO] loading encodings + face detector...
[INFO] starting video stream...
2024-08-05 17:04:03.321 WARNING streamlit.runtime.state.session_state_proxy: Ses
sion state does not function when running a script without `streamlit run`
qt.qpa.plugin: Could not find the Qt platform plugin "wayland" in "/home/pi/fyp-
env/lib/python3.11/site-packages/cv2/qt/plugins"
Engine response: Access denied. System locked.
2024-08-05 17:04:08.537
  Warning: to view this Streamlit app on a browser, run it with the following
  command:

    streamlit run homeai.py [ARGUMENTS]
```

Figure 6.4: System Lock and Access Denied by System

6.3 Testing Function of Smart Home Using Voice

After the system detect the authorized user, the system unlocks the function of the smart home. For this product all the function of smart home for the system and user uses the voice command where the user just use the voice command, and process the user request to activate the function of the virtual home assistant. The system focus for the user request to lock and unlock the door and play the music to the user.

6.3.1 Testing for Lock and Unlock the Door

When the user says the query for lock and unlock the door, the system try to process the user's request based on the keyword that has been set in the coding. For the lock the door, the keyword that has been set is "lock". When the system catches the keyword "lock" in the user query the system must response to the user that the system needs to lock the door.

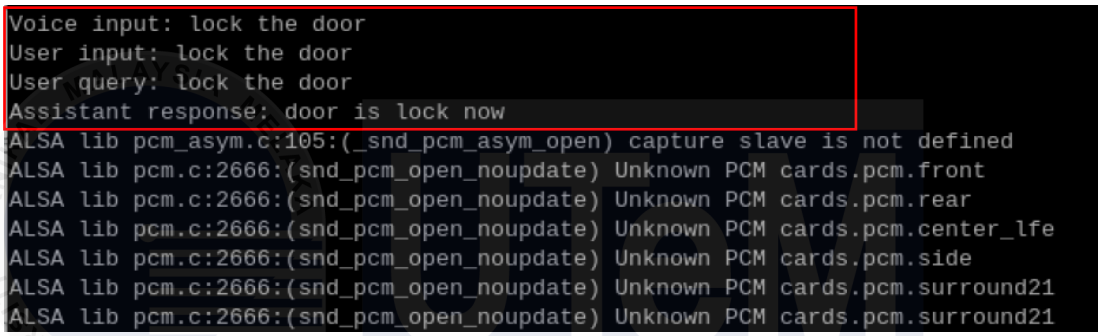

Figure 6.5: Door Lock Testing

Based in the figure 6.5 show the output when the system detects the keyword of the "lock". User query is what user ask for to the system and the assistant response is the system to response to the user by using voice and show the output. Figure 6.6 below also show the output when the system detects the keyword of "unlock" for the system to response for process the unlock the door and it response by using voice and show the output system.

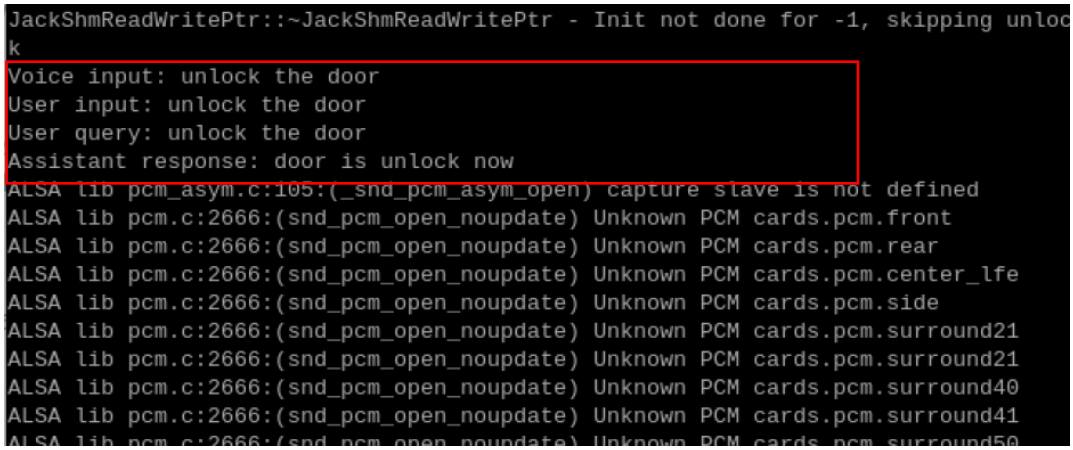

Figure 6.6: Door Unlock Testing

6.3.2 Playing the song

To test the system to play the song, user required to say the keyword that can activate the song in the system. For this system the keyword to wakeup the function of the virtual home assistant to play the song is “party”. For the example when the user says let’s party then the system would play the song that has been set. Then to stop the song the user just says to the system “stop the song” then the system would stop the song. The user can play and stop the sing just using the voice command.

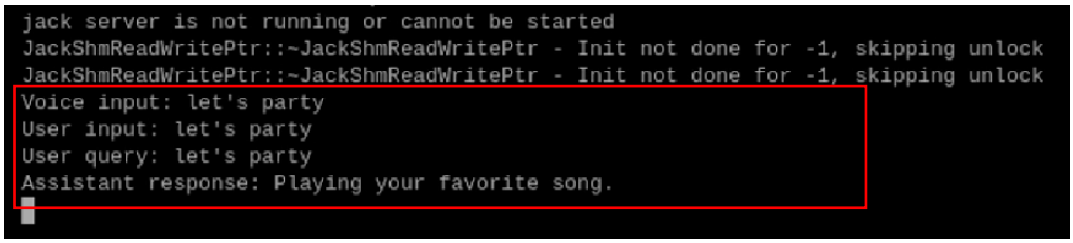

Figure 6.7: Testing for Playing Song

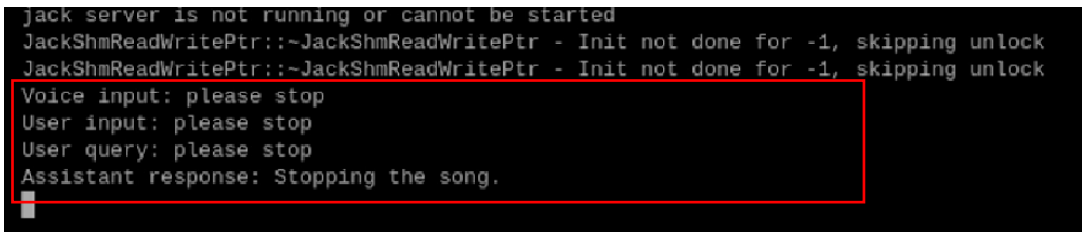

Figure 6.8: Testing for Stopping the Song

6.3.3 Ask for The Date Today

The virtual home assistant that has been developed also has the feature that can give the user for the current date. If the user's query has the keyword "date" then the system must process the query to give the user for the current date, and it response "the today's date is" then follow by the current date.

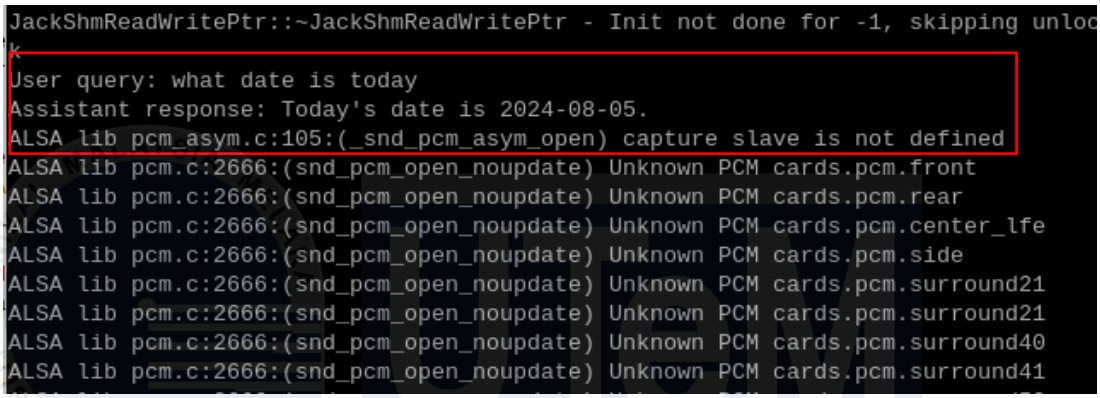

Figure 6.9: Testing for User Ask the Date

Test	Test Objective	Test Setup	Expected Result
Door Lock Testing	To test the user's voice command for lock the door	The user will use the voice to ask the system to lock the door	When the system detects the keyword "lock" in the user query it will response to the user that the door will lock.
Door Unlock Testing	To test the user's voice command for unlock the door	The user will use the voice to ask the system to unlock the door	When the system detects the keyword "unlock" in the user query it will response to the user that the door will unlock.

Playing the song	To test the user's voice command for play the song	The user will use the voice to ask the system to play the song for the user	When the system detects the keyword "song" in the user query it will response to the user "playing the song" and it will play the song.
Ask for the date today	To test the user's voice command for ask the today's date	The user will use the voice to ask the system to get the today's date	When the system detects the keyword "date" in the user query it will response "today's date is" and give the current date.

Table 6.2: Testing Function of Smart Home Using Voice

6.4 Testing Virtual Home Assistant F.F.I.F.A

The next feature that this virtual home assistant that called F.F.I.F.A have is it can provide the user for the information that need by user by connect the F.F.I.F.A to the google server by using the google Gemini API key. When the user asks the virtual assistant about something that need to get the information from the google, the virtual assistant would catch the full user query and covert to the text. Then virtual assistant connect to the google and get the information then convert it to the speech to response the user's query by using voice. This virtual assistant also has the error handling where if the virtual assistant doesn't catch the user query or if there have something that failure to the speech recognition, the system would notify the user. Beside that this F.F.I.F.A also have the feature that the user can close the virtual assistant by say the keyword to end the process of virtual assistant.

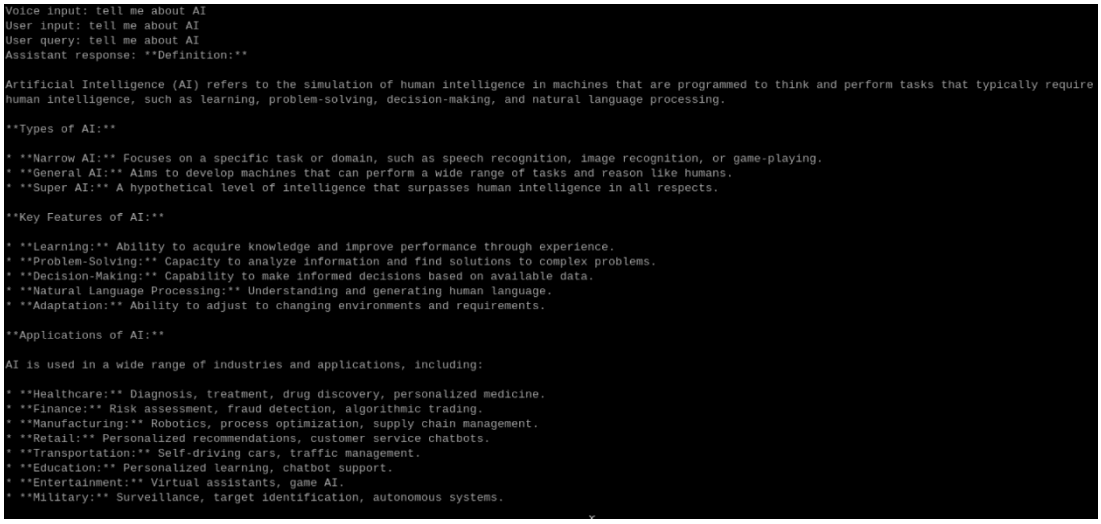

Figure 6.10: Testing for F.F.I.F.A to Get Information About AI

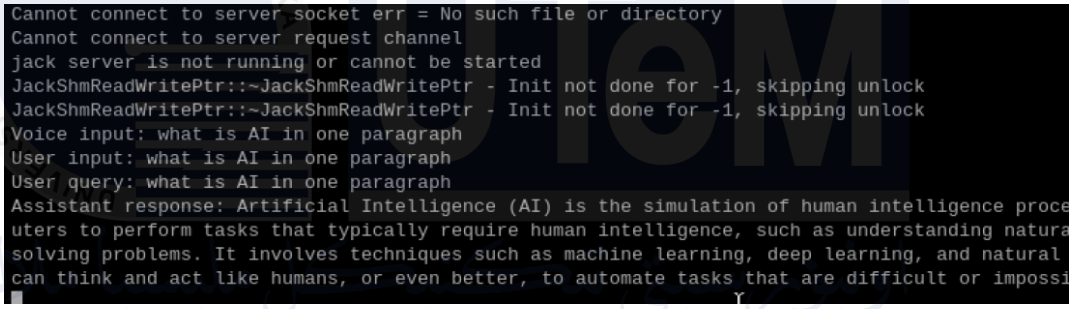

Figure 6.11: Testing for F.F.I.F.A to Get Information About AI

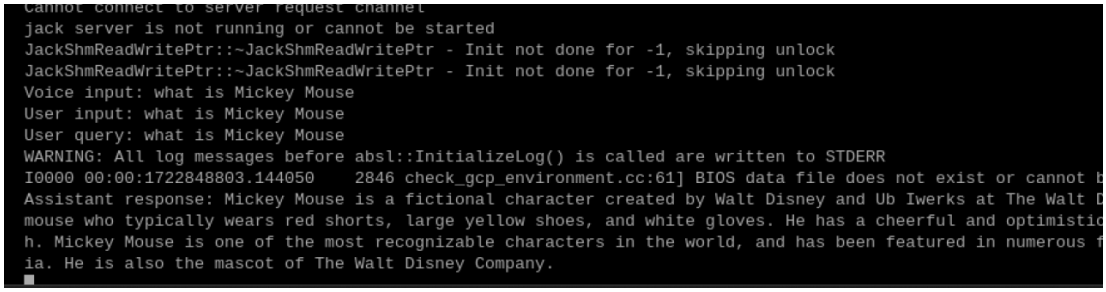

Figure 6.12: Testing for F.F.I.F.A to Get Information About Mickey Mouse

Based on figure 6.10, figure 6.11 and figure 6.12 show that the user asks for the information that require the system to get the information from the google engine and the system will response back to the user in the speech. This happen because the system has been connected to the google engine server by using the API key.

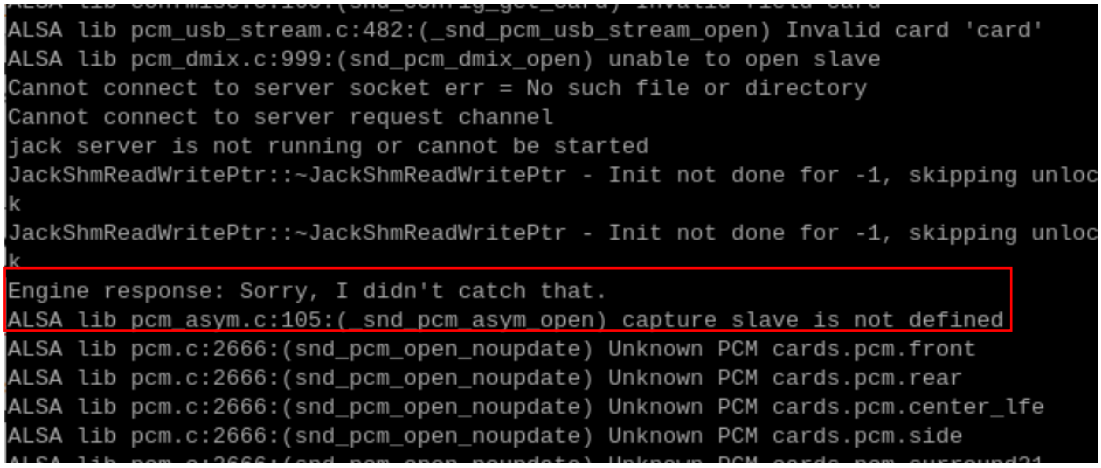

Figure 6.13: Testing for F.F.I.F.A Error Handling

When there is something that the system cannot catch the user query, the system would inform the user that the system cannot catch the user query and listen to the user query once again to get the query. The figure 6.13 show that how the error handling if the system cannot catch what the user says and show that the error handling for this system run correctly.

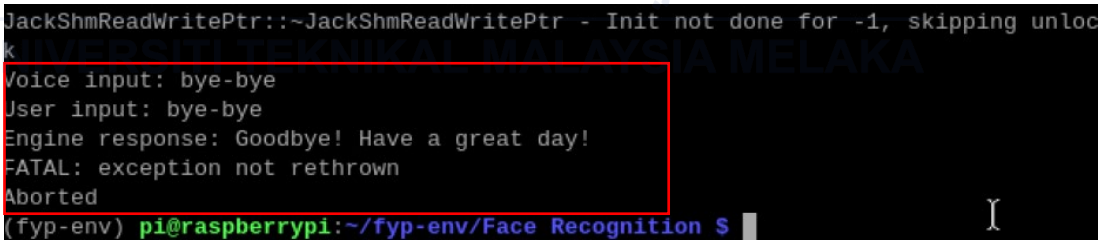

Figure 6.14: Testing for Ending the F.F.I.F.A function

If the user doesn't want to use the virtual home assistant, the system can be turn off and end their function by use the keyword to turn off. For this system the user just needs to say "bye" keyword and the system would detect the keyword and process to turn off the virtual assistant and response to the user by saying "Goodbye! Have a great day". The testing for the turn off the virtual assistant is show in figure 6.14.

6.5 Evaluation of F.F.I.F.A with Other Virtual Assistant

After the system has been developed and tested, then the system would be evaluated with the other virtual assistant to make sure that this system can be relevant with the other virtual assistant. In this part the F.F.I.F.A virtual assistant would compare with the Siri and Alexa.

Category	F.F.I.F.A	Siri	Alexa
Understanding Queries	Average ability to understanding user query	Strong ability to understanding user query (Emily Couvillon Alagha, 1 Nov 2019)	Struggled with understanding user query (Emily Couvillon Alagha, 1 Nov 2019)
Source of information	Provide answers with authoritative source because connect to google engine server	Provide answers with authoritative sources (Emily Couvillon Alagha, 1 Nov 2019)	Provide answers without authoritative sources (Emily Couvillon Alagha, 1 Nov 2019)
Implication for users	Average response	High-quality responses (Emily Couvillon Alagha, 1 Nov 2019)	Low reliability (Emily Couvillon Alagha, 1 Nov 2019)
User Experience	Less humanization but better natural to user	Less humanization, not natural to user (Ana Berdasco, 20 Nov 2019)	More humanization, more natural to user (Ana Berdasco, 20 Nov 2019)

Statistical Finding	Better performance	Lower performance (Ana Berdasco, 20 Nov 2019)	Better performance (Ana Berdasco, 20 Nov 2019)
User rating	Excellent	Below average (Ana Berdasco, 20 Nov 2019)	More excellent (Ana Berdasco, 20 Nov 2019)

Table 6.3: Evaluation of F.F.I.F.A with Other Virtual Assistant

Baes on the table 6.3 can show that the virtual assistant F.F.I.F.A can be relevant along with the other virtual assistant that has been established in the market. This because although the F.F.I.F.A is new in the development but can be compete and the function of the virtual assistant that provide can come with the same output to the other virtual assistant. Based on this comparison can show that in the future this F.F.I.F.A can be upgrade the function and the experience of user can be improved to be more humanization.

6.6 Conclusion

In conclusion, the testing phase for the Virtual Home Assistant Using Face Recognition has been tested for the functionality of the virtual assistant. It started with the user verification for the activate the system by using the face recognition, then tested for the function of the smart home that implement in this system then tested for the virtual assistant that can interact and give the information based on the user query by using the voice command and speech recognition. Then in this testing, the system has been compared with the other virtual assistant that has been implement and publish in the market. For the future improvement, this system can be upgrade for more

function of the virtual home assistant and can be more humanization to improve the user experience.

Chapter 7

Project Conclusion

7.1 Introduction

In this chapter, would discuss about the project's conclusion where include the summary for the entire project. this chapter include the project's strengths and weakness, contribution and the limitation for this project. For the project contribution will discuss about the process of the develop the system and the challenges that encountered during the development process. Finally, this chapter will discuss about the potential future work for the further enhance to the system that can be improve for the better performance.

7.2 Project Summarization

7.2.1 Project Objectives

When developing this project, all the project objective that has been stated on the Chapter 1 have been achieved. The first objective for this project that has been achieved is to analyse the user authentication for the secure smart home. This has been done started in Chapter 4 where the algorithm for the face recognition would be finalize and based on the evaluation on the chapter 2 this project use the OpenCV library for the face recognition where this library can give the accurate face recognition result. Then the code for the develop the face recognition would be discuss at the Chapter 5 for the step to develop the face recognition to recognize the user face. The first step is by installing the library that required for this system. Then the system capture the user face to collect the dataset then train the dataset. Then the result of the face recognition has been tested on the Chapter 6 where the system can recognize the

different user and if it detects the face that has been not register, the system would detect as the unknow user.

For the second objective that has been achieve is to develop the secure virtual assistant for the smart home by using the face recognition. This objective has been proven by integrate the virtual home assistant with the face recognition. When the system detects the authorized user that has been in the dataset it would unlock the virtual assistant, and the user now can use the function of the virtual assistant such as play and stop the song and lock or unlock the door. This system also can provide the information based on the user query where the information were get from the google engine server.

For the last objective is evaluate the function of the smart home and virtual assistant by using the user's voice. This can be referred to the Chapter 6 where in the testing phase this virtual home assistant has been compared to the virtual assistant that common use in the industries where the system has been compared to the Siri and Alexa. Based on the comparison, can conclude that this virtual home assistant can compete with the common virtual assistant where some of the function and category from either Siri or Alexa can be compete and the some of their weakness can be resolve in this virtual assistant. Overall, this project has been passes for the user experience and can be improve the function for better user experience.

7.2.2 Strengths and Weakness

For this project there have several strengths where the one of them is this virtual home assistant only can be access the function by the authorized user where it allows the authorized user to access the function of the home assistant and also can interact with the user by using the voice. If this project detects the unauthorized user

it would say the unknow user is detect and the system would block and shut down. This is the security method to make sure that the virtual home assistant can be only use by the user and prevent from the unwanted attack. The other strength for the system is the information that provide to the user that get from the google engine server. The speech recognition that implements in this system also have the better performance and it doesn't have problem to understanding user query.

Although this system has the strength, it also has the weakness that can be improve in the feature. The weakness for this project is the user verification may have challenges due to the different lighting conditions. The brightness and shadows may impact the effectiveness of the recognition. Furthermore, the user must speak the word clearly and slow to make sure that the system can catch the user query more accurate and give the better user experience. Moreover, if the system gets the information that have the special symbol, the system would speak with the incomprehensible words.

7.3 Project Contribution

The virtual home assistant using face recognition makes the big enhance in the virtual home assistant by make the system to integrate with the face recognition to make the better improvement in the security method. This integration not only makes the system safer, but it gives the user more flexibility and can improve the user experience by giving this the user the safe guaranty to use the system. face recognition is the easy way to use for the user authentication because it more friendly to the user and to the disability user. Furthermore, this system just needs the user to use the voice command also can make the better upgrade in the user experience. This system also come with the speech recognition and also can convert the text to audio to make sure the communication between the user and system more friendly and easy to use.

7.4 Project Limitation

For the first project limitation is the accuracy of the face recognition under the light condition. The brightness of the surrounding condition and the shadow disturbance would affect the face recognition. It may can the system does not recognize the authorized user or maybe some of other cases the system may can recognize the user as the wrong person. This system also needs the user to say the query clear and slowly to make it process more efficient. Sometime the system cannot capture what the user says and sometime the system also response to the user query with the wrong information because it cannot capture it clearly. The most important limitation for this system that need to be focus is the account that use for the API key where the account that provide the API key to this system need to pay for the monthly billing to make sure that the virtual home assistant can continuously connect to the google engine server.

7.5 Future Works

In the future, this project can make the future update and improvement to increase the user experience and also make this project more reliable in the market and can compete with the current virtual assistant that common use in the daily life. The enhancement in the future could be benefit in several areas:

i. Enhanced in the functionality: The system can be considered to add more functionality of the smart home to make the verity of the function to make sure that the user experience can be increase.

ii. Enhanced facial recognition: For the face recognition system can be enhanced by add the emotion recognition by detect the mood of the user's face. This can increase the security method because if the user faces the emergency that

cannot require the user's voice, the system can detect the face mood and can give the help to the user.

iii. Two-factor authentication: by add the two-factor authentication to the virtual home assistant F.F.I.F.A for the increase the security method in this system. This can be done by add the other user authentication such as the passcode or add the voice recognition to make sure that the system only can be access by the authorized user.

iv. Enhanced in humanization: this system can be enhanced by upgrade it to be more human like speech pattern where the keyword of the user query and user command more like the natural speech pattern. The voice of the virtual home assistant also can be enhanced to make it more like the human voice. This can be improving the user experience and make the user enjoy to use the system.

7.6 Conclusion

This system has been successfully designed, developed and tested for the Virtual Home Assistant Using Face Recognition. The project methodology that use for this project consisted of the process analysis and design, develop, demonstrate and review, testing and implementation. The development processes the system has been meet the project objective and in the testing phase the system has been evaluated for the performance of the function, the accuracy of the face recognition and also the interaction of the user and the system. The user interaction with the system has reached a satisfactory level but it can be improved in the future to make sura that the user can satisfy when they use this system. The face recognition that implements in this system has meet the best performance where the speed of the recognition is in the good performance. This project is developed to focus for the user-friendly system and make it more secure when the user use this system.

References

Lajos, Hanzo. (2023). Voice Recognition System for Desktop Assistant. Advances in intelligent systems and computing, doi: 10.1007/978-981-19-9819-5_48

(2023). Multipurpose Virtual Assistant Using Machine Learning. Proceeding international conference on science and engineering, doi: 10.52783/cienceng.v11i1.126

Ms., Shruti, Sarwate., Harshal, N, Suryawanshi., Rutvik, Borade., Rushikesh, Farkase., Yash, Selukar., G, D, Indurkar. (2022). Personal Virtual Assistant ("DANI"). International journal of innovations in engineering and science, doi: 10.46335/ijies.2022.7.8.14

(2023). Data Security Threats On Smart Devices At Home. ARPHA Conference Abstracts, doi: 10.3897/aca.6.e106978

Smart Home Is Not Smart Enough to Protect You - Protocols, Challenges and Open Issues.

https://www.sciencedirect.com/science/article/pii/S1877050919316680

Hasibul, Alam., Emmett, Tomai. (2023). Security Attacks and Countermeasures in Smart Homes. doi: 10.5121/ijci.2023.120209

Ondra, Eka, Putra., Retno, Devita. (2023). Safe Security System Using Face Recognition Based on IoT. Sinkron : jurnal dan penelitian teknik informatika, doi: 10.33395/sinkron.v8i2.12231

(2023). Enhancing Smart Home Security with Face Recognition using Deep Learning. International Journal For Science Technology And Engineering, doi: 10.22214/ijraset.2023.50243

A., Rahim., Yanru, Zhong., Tariq, Ahmad. (2022). A Deep Learning-Based Intelligent Face Recognition Method in the Internet of Home Things for Security

Gesha, Warilotte, Erwinda., Suryo, Adi, Wibowo., Deddy, Rudhistiar. (2023). Implementasi face recognition dan rfid sebagai fitur security pada smart home. JATI (Jurnal Mahasiswa Teknik Informatika), doi: 10.36040/jati.v6i2.5350

Thai, Viet, Dang. (2022). Smart home Management System with Face Recognition Based on ArcFace Model in Deep Convolutional Neural Network. Journal of Robotics and Control (JRC), doi: 10.18196/jrc.v3i6.15978

P, Srinivas., T, Sai, Teja., CH, Bhavana., R, Likhith., K, Sathish, Kuma. (2020). Raspberry Pi based personal voice assistant using python. doi: 10.33564/IJEAST.2020.V04I11.020

George, Vardakis., George, Tsamis., Eleftheria, Koutsaki., Kondylakis, Haridimos., Nikos, Papadakis. (2022). Smart Home: Deep Learning as a Method for Machine Learning in Recognition of Face, Silhouette and Human Activity in the Service of a Safe Home. Electronics, doi: 10.3390/electronics11101622

Emily, Couvillon, Alagha., Rachel, Renee, Helbing. (2019). (8) Evaluating the quality of voice assistants' responses to consumer health questions about vaccines: an exploratory comparison of Alexa, Google Assistant and Siri.. doi: 10.1136/BMJHCI-2019-100075

Ana, Berdasco., Gustavo, López., Ignacio, Díaz-Oreiro., Luis, Quesada., Luis, A., Guerrero. (2019). (2) User Experience Comparison of Intelligent Personal Assistants: Alexa, Google Assistant, Siri and Cortana.. doi: 10.3390/PROCEEDINGS2019031051