DEVELOPMENT OF A MINIATURE SMART HOME TESTBED FOR RESEARCH AND EDUCATION.

By

TRUNG NGUYEN

Bachelor of Science
in Electronics and Telecommunications Engineering
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
2014

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 2017

DEVELOPMENT OF A MINIATURE SMART HOME TESTBED

FOR RESEARCH AND EDUCATION.

Thesis Approved:

Dr. Weihua Sheng

Thesis Advisor

Dr. Qi Cheng

Dr. Yanmin Gong

ACKNOWLEDGMENTS

I would first like to thank my thesis advisor Dr. Weihua Sheng of the School of Electrical and Computer Engineering at Oklahoma State University. The door to Prof. Sheng office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this work to be my own work, but steered me in the right direction whenever he thought I needed it.

I would also like to thank the experts who were involved in the validation survey for this research project: Mr. Barth Lakshmanan, Mr. Minh Pham, Mr. Ha Do, Mr. Chengjie Lin. Without their passionate participation and input, the validation survey could not have been successfully conducted.

I would also like to acknowledge Dr. Qi Cheng, Dr. Gong Yanmin and Dr. Jingtong Hu as the second reader of this thesis, and I am gratefully indebted to them for their very valuable comments on this thesis.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Trung Nguyen[1]

[1]Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University

Name: Trung Nguyen Date of Degree: July, 2017

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: DEVELOPMENT OF A MINIATURE SMART HOME TESTBED FOR RESEARCH AND EDUCATION.

Candidate for the Degree of Master of Science

Major Field: Electrical and Computer Engineering

We present a cloud based smart home testbed using a miniature doll house. In particular, we use this testbed to demonstrate the energy conservation system with the interconnected network of devices, the robust security system with face recognition feature, and the home assistant robot with voice-controller. The test bed consists of a smart home server, a home controller, a smart home assistant robot, a security camera, and appliances communicating with each other using web-socket and existing Social Network Software SDKs. The proposed testbed allows research and education in areas such as smart-grid, wireless sensor networks, machine learning, pattern recognition, embedded programming, natural language processing, social media sharing etc. We also propose a security system based on face recognition. In particular, we develop this system for giving access into a home for authenticated people. The classifier is trained using a new adaptive learning method. The training data is initially collected from social networks and the accuracy of the classifier is incrementally improved as the user starts using this system. A novel method has been introduced to improve the classifier model by human interaction and social media. By using a deep learning framework - TensorFlow, it will be easy to reuse the framework to adapt with many devices and applications.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In this chapter, we present about motivations and related works in the domain of smart home systems, smart home energy systems, security systems, voice-controlled systems and smart home assistant robots.

1.1 Motivations

Nowadays, smart homes extend not only to customers who enjoy leisure and comfort, but also to many researchers in a wide variety of research areas and fields. A typical smart home features seamlessly integrated security systems, air conditioning systems, entertainment systems, lighting systems, and smart appliances. Wireless technologies connect these systems together. However, these interconnected systems require synergy among them to make a smart home environment. An example of smart home systems is described by Kuzlu et al. [1] With those technologies, a smart home should:

- Intelligently allocate home power supply based on demands of home appliances, while directing excessive energy back to the grid or local storage batteries;
- Incorporate energy-storage systems to stabilize energy flow;
- Capture data related to power consumption of individual appliances;
- Use an intelligent core to provide grounds for synergy among central computing systems and all terminations, much like a human brain;
- Intelligently change the environment based on human activities or behaviors; [2]

- Include an artificial intelligence assistant robot which controls smart home appliances by voice; [3] [4]

- Monitor the entirely the smart home with security cameras which detect and recognize person's faces.

Even though the smart home industry is relatively new, it is rapidly growing. Companies like Microsoft, Samsung, Google, Apple and many startup companies are already sold many smart home devices [5].

It is important to develop a research and education testbed which can be used as a platform to train the next generation of engineers in smart home technologies. We propose a smart home testbed using a miniature doll house which imitates a smart home. [6] A number of sensors and devices are utilized in the mini house to simulate the actual smart home environment. The sensors and appliances are connected to a home controller which communicates with a central management smart server using web sockets. The smart home server communicates in the same way with other clients such as a smart phone, a smart home assistant robot, etc. With this proposed framework, we show that energy can be conserved and utilized based on the demand.

Also, we describe two important features of the smart home, which are the security system and the smart home assistant robot. Security systems have a crucial role to safeguard people. It is necessary to have a robust system which can distinguish between people and respond differently based on their privileges. It is essential that the system is robust to recognize people and that the training should be accomplished without much difficulty. By using a novel machine learning framework TensorFlow [7], we can take an advantage of the state-of-art recognizers using convolutional neural networks (CNN) [8] which outperform the human's recognition rate. Moreover, voice controlled systems take an important role in the smart home and currently become a hot field with devices like Amazon Echo [9] and Google Home [10], but those are private and paid services. By using an AI voice controlled system MyCroft [11], we implemented and adapted it to our smart home system with an impressive performance. This thesis also aims at providing an interactive feature to the smart home system that analyzes human voice and responds to the queries. As an extension to the voice controlled home automation, this system tries to address all the queries, thereby providing a certain amount of information and entertainment. We can get the basic information on various things such as time, weather, distance and entertainment like jokes and songs. The system has the main unit which is the Raspberry Pi for voice recognition, along with the Google API [12] and MyCroft [11] for voice synthesis. The system analyzes the human voice command, converts it into text, processes the query and converts the answer back to speech, which is the response of the system. The thesis gives a detailed explanation of the development of the Mycroft service.

We also built a 3D-printed BI robot which includes a camera, a Raspberry Pi, a speaker, a microphone, a screen to be an assistant robot with the voice controller, Wechat controller, and show and tell feature.

1.2 Related Works

In this section, we briefly review the existing smart home projects in terms of smart home service systems, smart home energy systems, smart home security systems and voice controlled systems.

The smart home project in the paper [13] focused on the design of a smart home energy management system based on ZigBee protocol with the support of sensor networks and actuator components. The smart home in the paper [14] dealt with processing, networking abilities, and transferring signals from device to device and from device to server. Kumar et al. [15] focused on building a flexible smart home system which used the web sockets protocol to broadcast and subscribe information between an Android App and an Arduino web server.

Angelo et al. [16] developed a decision-support algorithms. The user first assign values to desired energy services and then scheduling their available distributed energy resource (DER) to maximize net benefits. Also, Hu et al. [17] proposed a smart home energy management system where consumers can easily access a real-time, price-responsive data for a heater, an air conditioner, and an electric vehicle, and it used a machine learning algorithm to help consumers reduce their electricity bills. Jahn et al. [18] presented a smart home system with energy efficiency features over a peer to peer (P2P) network to calculate and analyze the consumed energy at a device level. Farah et al. [9] introduced a smart system with smart home applications such as a temperature monitoring feature and a security feature with temperature and motion sensors.

In addition, in terms of security system and face recognition, the state of art of face recognition reaches the human performance with robust results. Various home security systems have been used in the market in many big companies such as ADT [19], Vivint [20], and Protect America [21]. However, none of them has the face-recognition feature in their systems because of low confidence performances and high computational demand. With the rapid development of smart devices, Netatmo [22] presented a device using a deep neural network to recognize faces, but their system is still far from perfect since the camera is slow. The live feed lags, the notifications are delayed, and it takes a while to learn faces.

The robustness of face recognition systems depends on the changes in light conditions or expressions. Several papers have proposed various techniques for face recognition under those conditions [23]. The Eigenfaces method [24] described variant extracted feature to address the above factors. The Facenet in paper [25] used a deep convolutional neural network with the architecture of the Inception model [26] from Google and used a novel online triplet mining method to train an intermediate bottleneck layer. On the widely used Labeled Faces in the Wild (LFW) dataset [27], the Facenet system achieved a new record accuracy of 99.63%. However, unfortunately, not only the size of the database increases when we have more data, but also its computational cost increases and the recognition accuracy declines accordingly. That is why incremental learning is a learning algorithm which approaches to handle large-scale training data for a better efficiency and accuracy. A brief definition of incremental learning is that learning is a gradual process with new data. The existed classifier layer are identified with the new classes [28] [29]. Besides, while many incremental learning works focused more on efficiency than accuracy, Bengio et al. presented a deeper vision that improves speed and quality if increments are made properly, which is called curriculum learning [30]. Its key idea is to begin learning on low-resolution images and then gradually increase to high-resolution image [31]. That is why we use 96x96 pixels cropped as the input data which is also mentioned in the Facenet paper [25] for the best performance training data size. We also describe a novel method to collect data from social media by using Facebook API [32] and ask people to label the unidentified faces, that supplies the neural network model with new training data. The interface is also designed with easy uses in many types of devices.

Regarding to voice controlled system, with rapidly growing technology, all the leading tech companies are in a race to capture the market through the voice controlled gadgets that can provide the user with a smart, interactive feature by processing natural voice without actually touching the device such as Siri (Apple), Amazon Echo [33], Google Assistant, etc. Zhu et al. presented a voice control system for a home automation network [34] based on ZigBee networks and Speaker Independent Automatic Speech Recognition (SI-ASR). Guo et al. described an interactive voice-controller which applied to home automation [35]. However, those voice controlled systems are in poor performance and only work with limited commands. By taking the advantage of AI, we built an intelligent voice controlled system based on Mycroft [11] and Google API [12].

Inspired by the research presented by Srepan et al. [36], this thesis aims to efficiently distribute home control tasks and automatically manage energy consumption in a miniature smart home. The proposed configuration and management of the smart home are more convenient and efficient by taking advantage of the connection between a smart home server and a home controller. Users can manage and configure smart home appliances via a smart phone application, a web browser, or a smart home assistant robot. This smart home testbed can serve as a platform for research and education in several areas including Internet of Things, Artificial Intelligence, social networks, cloud computing, computer networking, etc.

An overview of the rest of the thesis is as follows: in Chapter 2, we will describe overview of our miniature smart home system. The smart home management system is discussed in Chapter 3; following is the smart home security system in Chapter 4. Our smart home assistant robot is described in Chapter 5. Finally, Chapter 6 gives the conclusions and future works.

CHAPTER 2

SMART HOME SYSTEM

In this chapter, we introduce the miniature smart home testbed. This chapter will also give an idea of how the smart home system looks like, what devices are inside it, and how it works.

2.1 Smart Home Architecture

This section describes the setup of our smart home testbed which consists of a smart home server on a private cloud, a home controller, a smart home assistant robot, a smart security system, a public cloud, appliances, and clients interacting with the users as shown in Figure 2.1.

Each client node can communicate and control smart home appliances by connecting to the private cloud which runs the smart home server. The architecture is designed, so that failure of one particular client node will not blackout the entire system. For example, the Raspberry Pi home controller connects to various sensors and actuators in the miniature smart home. It receives data from different sensors, and it can make decisions based on the usage and availability of resources to control the appliances. Afterward, the home controller also updates its decision to the smart home server which in turn reflects the change to all connected clients. We make use of web sockets provided by Socket.IO to implement this feature. Whenever the smart home server receives an update, it will send a broadcast message to all the clients to reflect the changes.

The smart home assistant robot includes a voice controller and a Wechat con-

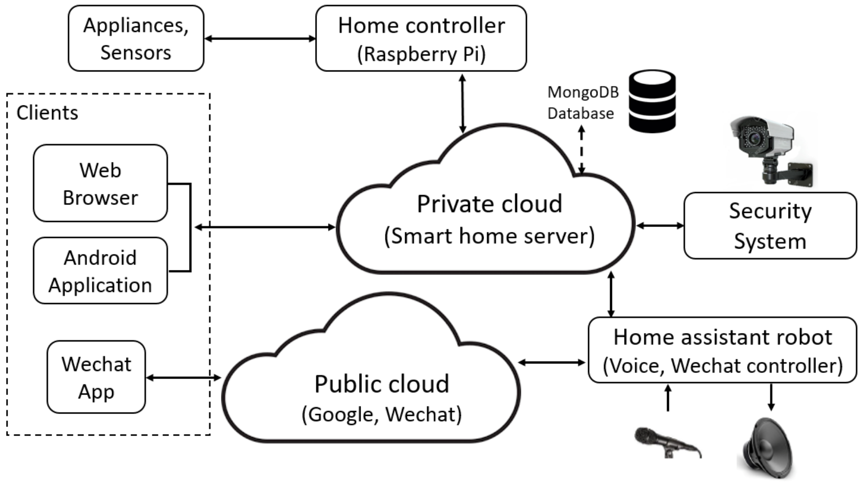

Figure 2.1: Smart Home System Architecture.

troller. The voice controller uses a person's voice to control smart home appliances. It also enables questions and answers from MyCroft skills [11]. The Wechat controller uses the Wechat server [37] to control home appliances and query their status. Any number of clients can connect to the system such as a web browser, an Android app, a Wechat app, etc. This architecture offers flexibility to add more appliances such as cameras, sensors, lights, fans, etc. which can be controlled by using the provided application program interfaces (APIs).

2.2 Design of Miniature Smart Home

The proposed system includes a smart home server, a home controller, a smart home assistant robot, a public cloud, a security system, a solar panel, battery, appliances, and clients. Some of the main components are explained in detail in this section.

2.2.1 Home Controller

The Raspberry Pi home controller manages all sensors and actuators in the miniature smart home as shown in Figure 2.2. The control circuitry is connected to a touch display to show the appliance's power consumption, battery percentage, thermostat value, and other related information. A portable auto heater and defroster forming the air conditioning (AC) system is controlled based on the current temperature within the home and the set temperature. The control of the air conditioner is based on the mode in which the AC operates. There are three modes: off, heating and cooling, which are similar to the traditional thermostat. Different sensors such as temperature, current, and voltage sensors are connected to ADC (analog to digital converter) to get digital values. The miniature smart home testbed is sealed using acrylic glasses to maintain the temperature within the environment.

The smart home system is powered by renewable energy. The excess energy which is obtained from the renewable source is stored in a battery and is used when the renewable energy is not available. A 100 Watt, 12 V mono-crystalline solar panel and a 12 V rechargeable sealed lead acid are utilized in the miniature smart home testbed. The solar panel connects to the battery through a charge controller which regulates the input power to the battery. The charge controller can also tell when the battery is fully charged, when solar energy is not available, etc.

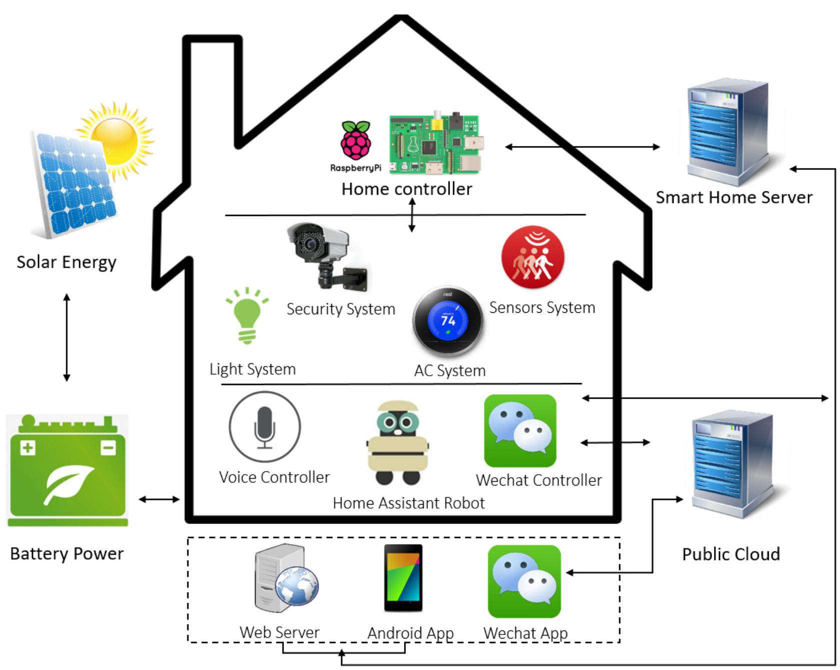

Figure 2.2: System Overview.

The current and voltage sensors are used to sense the power consumption of various appliances in the mini smart home. The home controller measures those values and also updates to the smart home server. This setup makes it possible to remotely monitor the instantaneous power produced by the solar panel and the power consumed by all the devices.

The lights and air conditioner are controlled by using MOSFETs and relays. The states of these appliances are controlled by various sensor values and user's requests. When the user sends a request to the smart home server, the server forwards the message to the home controller and sends an acknowledgment to the requested client.

The block diagram of the home controller is shown in Figure 2.3, which includes the hardware elements of the smart home and the connection between them. Also, the selected GPIO output pins between the Raspberry Pi and appliances are presented in the block diagram.

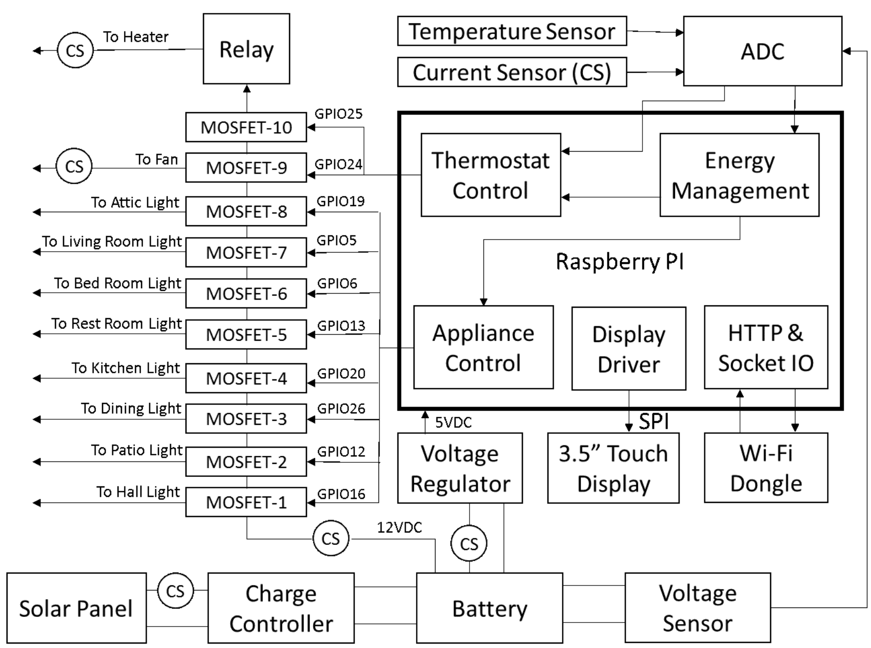

Figure 2.3: Block Diagram of The Home Controller.

2.2.2 Smart Home Server

The smart home server receives data from various clients interfaces, and it contains a program with predefined APIs. The state assigned by the server for any particular actuator can be accessed by the client using HTTP requests or web sockets as shown in Figure 2.4. The network framework includes a web browser (REST API and logging services), a MongoDB database, and a connection with the home controller.

The server needs to be logged using HTTP to obtain the instantaneous actuator's status of the smart home. Whenever a request is sent to the server, a new port/socket is opened, and the data is transferred through this port. Once the data has been transferred, the port closes. Further, it is not possible for the server to initiate communication with any client. Hence, the client has to request the server for updated information every few seconds. On the other hand, web sockets are the extension of traditional sockets. The server opens up the socket for clients and will be able to broadcast messages without any request from the clients thus avoiding the polling operation. The smart home server is hosted on the private cloud, and it is linked to a database. All the sensor and actuator values are stored in the MongoDB database [38]. Many clients can communicate with the server such as an Android app, a Wechat app, a web browser, etc. by using a set of predefined REST API [39].

To achieve even more features on the smart home server, we decided to use Home Assistant platform [40] on the smart home server, it is an open source home automation platform which includes many standard ways to connect with smart devices in the market such as Amazon Echo [33], NEST [41], etc.

The smart home server also supports many communication protocols such as MQTT [42], Zigbee [43], and Z-wave [44] which are three traditional protocols in smart home systems.

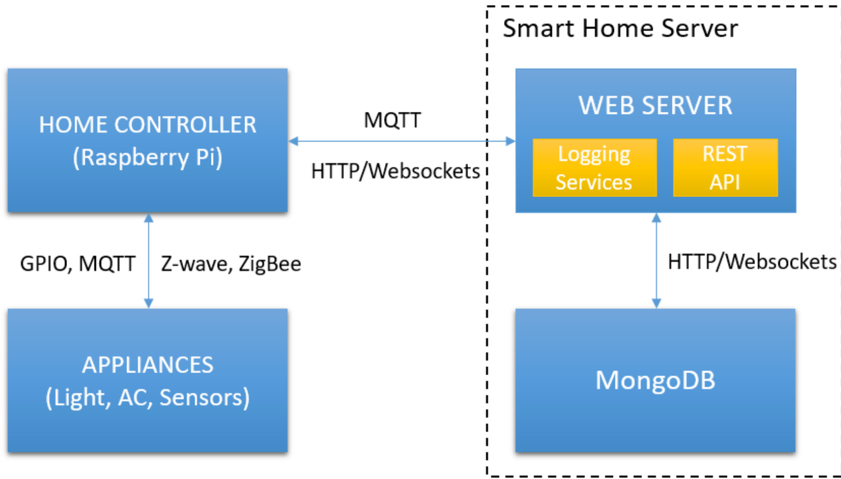

Figure 2.4: Smart Home Server Principle.

2.3 Experimental Results

2.3.1 Miniature Smart Home Testbed

The mini smart home testbed was implemented as shown in Figure 2.5. The setup was built through collaboration between undergraduate students and ASCC Lab members. It used the solar panel to generate the power and store to the battery.

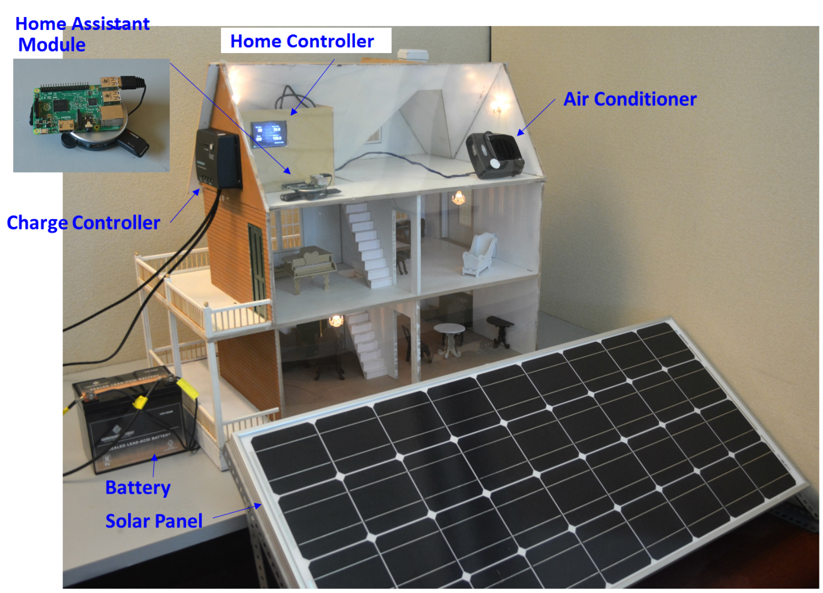

Figure 2.5: Smart Home Testbed System.

2.3.2 Stack Server

The smart home server is located on a stack server. The server includes three nodes: 1 controller node on the top and 2 compute nodes at the bottom. Our server includes Ubuntu, Cirros and CentOs operation environments and instances hosted by the QEMU hypervisor [45] on the compute nodes. The server can provide a capacity up to 12 VCPUs (time slot of the processor) with 16Gb Ram and 100Gb Root Disk. With high computational power, the server is suitable for deep neural network training and testing. The stack server is shown in Figure 2.6.

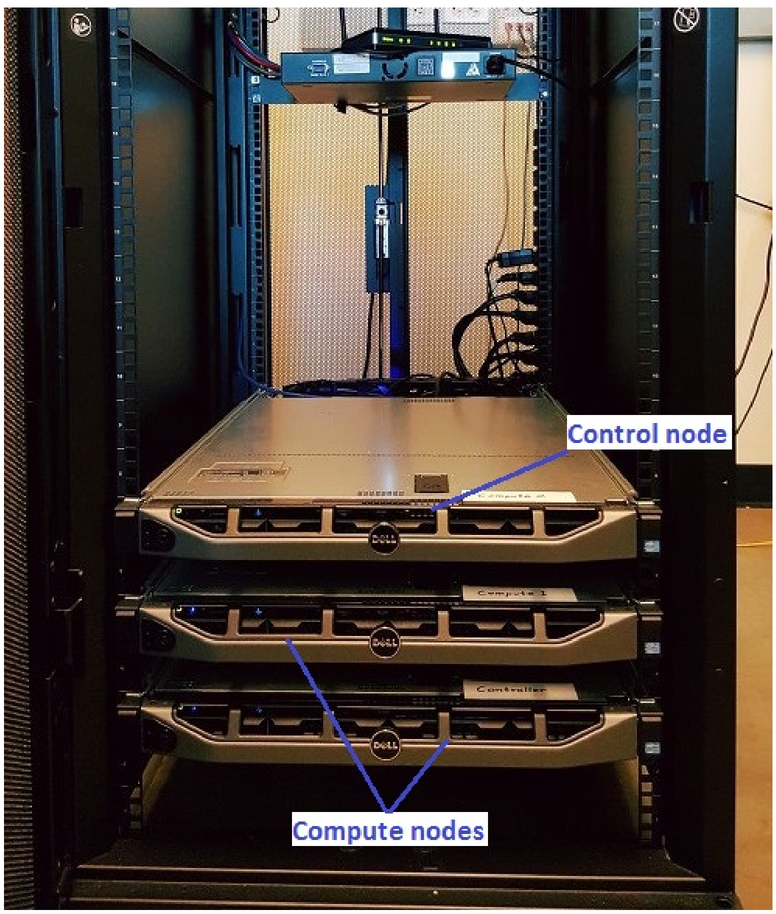

Figure 2.6: The stack server which hosts the Smart Home Server.

2.3.3 Client Interfaces

The Android application interface is shown in Figure 2.7 (left) with information of the device's status, control condition, temperature, and also device power consumption. There is also another way to connect with the smart home via the Wechat app as shown in Figure 2.7 (right). By sending a text or voice message to the Wechat website cloud, we can control the smart home system and also query the status of the smart home.

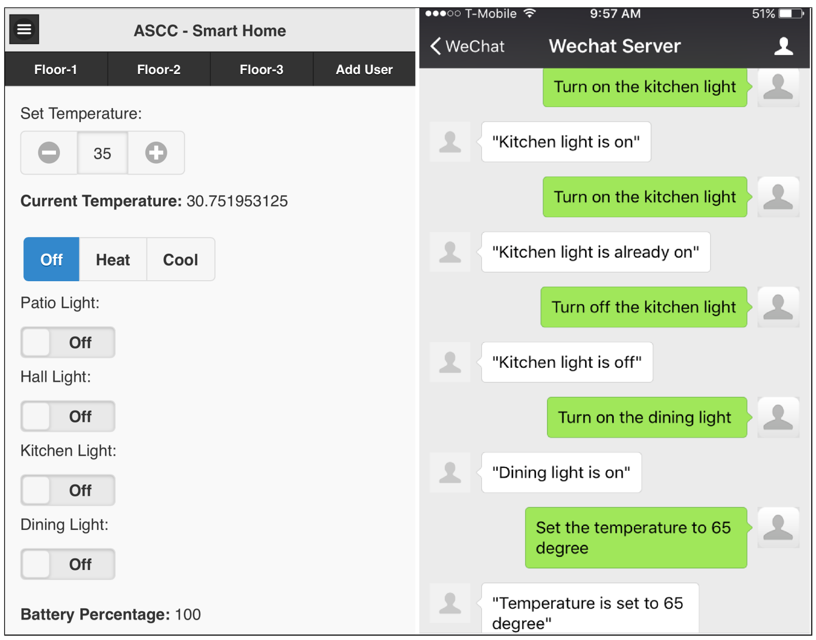

Figure 2.7: Smart Home Interfaces using the Android App (left) and the Wechat App (right).

CHAPTER 3

SMART HOME ENERGY MANAGEMENT SYSTEM

In this chapter, we will describe the smart home energy management system. This chapter will give a working flow of the energy management system and explain the algorithm to save more energy.

3.1 Energy Management System

3.1.1 Management System

The management system includes sensing interface, energy-aware, and services management. Figure 3.1 shows the energy management system.

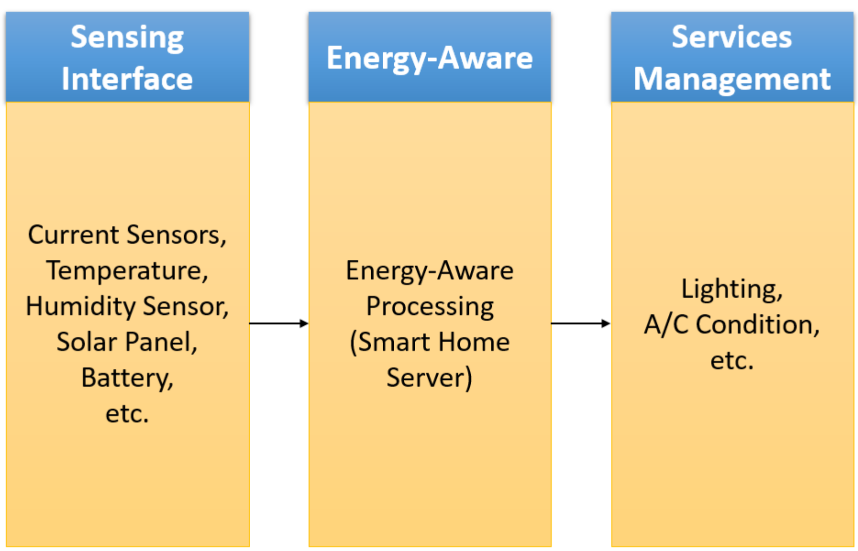

Figure 3.1: Energy Management System.

Sensing Interface

The sensing interface is designed to gather information from appliances such as consumed power, current sensors values, temperature values. By using current and voltage sensors with a wireless module (MQTT [42] and WiFi [43] protocol), our smart home is able to control and track the status of appliances from distance.

Energy-Aware

Then, the data is sent to the smart home server. Upon receiving the information, the energy-aware will analyze, process a command based on energy-aware services and send to the services management to control the energy mode of the system.

Services Management

After the decision was made, the energy-aware services from the smart home server will send the decision mode to the services management which includes devices and appliances such as lights and AC conditioner, etc. to control and save energy.

3.1.2 Energy-aware Services

One of the problems with the renewable energy, in particular solar energy, is that it cannot produce power continuously throughout the day. It depends on the weather and the energy generation, while power demand varies within a smart home environment. It is necessary to change loads based on availability and maximize the usage of available resources. On the other hand, instead of fully cutting down the usage, energy can be conserved by reducing the comfort level of the user. This approach to conserving energy is better as it does not cut down supply to any device rather it reduces the power consumed. With this idea, we propose a method to adjust the temperature level of the air conditioner based on the available energy resource. With the current and voltage sensors, we measure the instantaneous power consumed by the air conditioning system and lighting system. We also measure the power generated by the solar panel in a similar fashion. The battery percentage or the state-of-charge of the battery is determined by measuring the voltage. This method is simple, but it can be inaccurate because cell materials and temperature affect the voltage. The most blatant error of the voltage-based SoC occurs when disturbing a battery with a charge or discharge. The resulting agitation distorts the voltage and it no longer represents a correct SoC reference. To get accurate readings, the battery needs to rest in the open circuit. We use this method to measure the battery percentage along with the other measured values. A table for voltage vs SoC is shown in Figure 3.2.

Voltage	State of Charge
12.6+	100%
12.5	90%
12.42	80%
12.32	70%
12.20	60%
12.06	50%
11.9	40%
11.75	30%
11.58	20%
11.31	10%
10.5	0%

Figure 3.2: Battery Voltage vs State-of-charge.

We developed an energy saving method to prove that the interconnected devices can help conserve energy. By monitoring and controlling devices, we can turn off the unnecessary actuators anytime.

The room and set thermostat temperatures, the mode of the air conditioner and the battery percentage are displayed on the LCD screen which connects to the home controller. The current sensors are used to measure the output current from the solar panel, based on which we calculate the power consumed by the entire miniature smart home. By calculating these, we can know if the solar panel is supplying enough energy to power the smart home, or if the battery power is being consumed. Thus, if the power obtained from the solar panel is more than the power consumed, the battery is charged, and if the solar panel power is less than the power consumed, the battery is discharged. In a mathematical form, we can express it as follows:

- If Psolar - Pconsumed > 0 then battery is charging

- If Psolar - Pconsumed < 0 then battery is discharging

Our objective is to maximize the hours of usage of available energy. This should be achieved without cutting down any supply to the user. Hence, we define three modes in our system: normal, economy, and sleep.

Figure 3.3 shows the flowchart of controlling the lights and the air conditioner during critical conditions.

When an abundant amount of energy is available, i.e., when the state-of-charge of the battery is greater than 30% and the power produced is greater than or equal to the amount of power consumed by the smart home, then it will work in the normal mode in which none of the parameters in the system is affected. The entire system is under the control of the user and the system does not turn off any appliance. Ideally, the smart home system designed with a proper choice of renewable energy should operate in the normal mode for most of the time. However, there is a chance that the system cannot generate sufficient energy to meet the demand. In such a situation, the system automatically switches to the economy mode. This happens when the battery percentage is within 10-30% and the energy consumed is greater than the energy produced. When the system enters the economy mode it tries to change the parameters in the system such that the users can still use the system but at a reduced level. For example, when the system enters the economy mode, the system automatically changes the temperature threshold of the air conditioner. In the economy mode, the value of the set temperature is increased or decreased by 5 Celsius based on the environmental temperature. It provides a longer duty cycle and hence increases the efficiency of the system. The economy mode also reduces the power supplied to the lighting system by reducing the voltage level. It results in reduced illuminance inside the home. The economy mode basically tries to balance between the comfort level of the user and the power consumption. In the worst case, the user may run out of energy and would have very little energy left and the system will enter the sleep mode. This happens when the state-of-charge of the battery is less than 10% and the power consumed is greater than the power produced. When the system enters the sleep mode, the air conditioning system gets turned off and only the living room light gets the power supply. The system tries to save the remaining power for basic necessities such as charging a mobile phone.

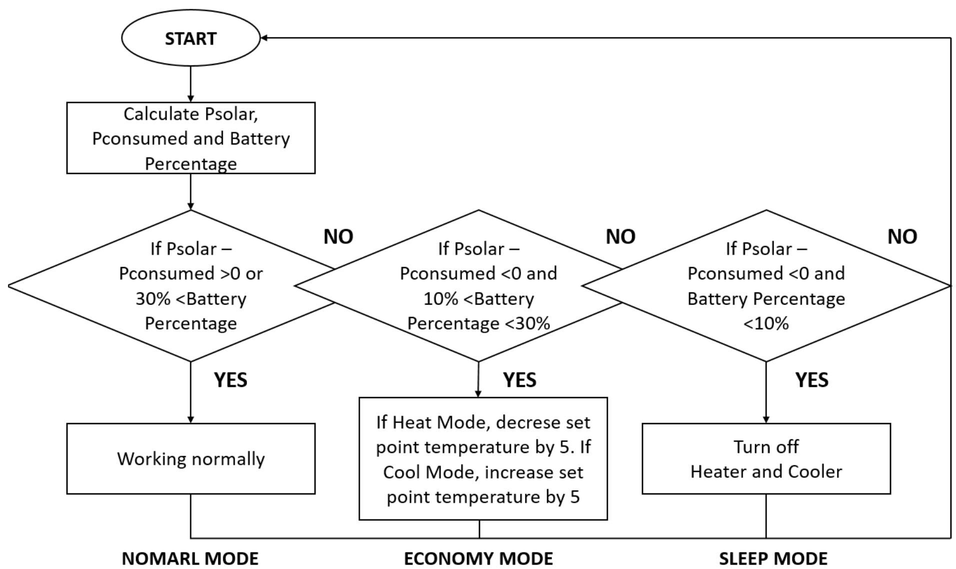

Figure 3.3: Energy Conservation Flowchart.

3.2 Experimental Results

3.2.1 Power Consumption

We ran the energy conservation program for one week. Some of components are used fixed current consumed value (light, ac conditioner, raspberry pi) and some have to be tracked in real-time (solar panel, battery). The power consumption is shown in Figure 3.4, and it contains the power consumption of all appliances and power sources of the smart home. The power consumption with the energy conservation method is lower than without applying this method. We expect that in a real system it would also save more power since all the appliances in the miniature smart home use a small amount of power.

Current Consumption

Device	Model	Active Current	Standby Current
Home Controller	Raspberry Pi	172 mA	50 mA
Temperature Sensor	TMP36	1 mA	750 nA
Light	Lightbulb	1 mA	0 mA
AC conditioning	RoadPro fan	500 mA	2 mA

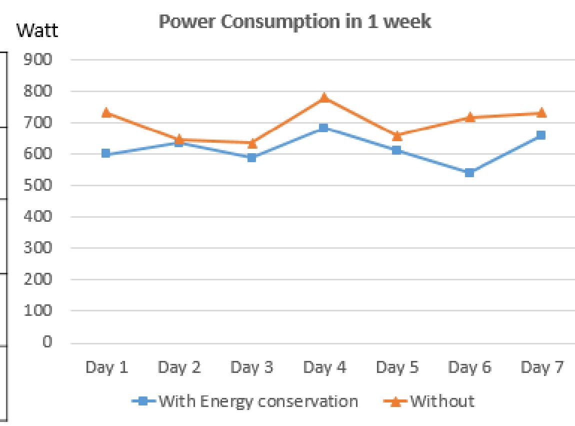

Figure 3.4: Current Consumption table (left) and Power Comparison chart (right)

CHAPTER 4

SMART HOME SECURITY SYSTEM

In this chapter, we will present about the smart home security system which is mainly smart camera and its interaction. The security system is an AI technology with high confident and auto learning and updating.

4.1 Security System with Face Recognition

This section describes our security system setup in the smart home. This is an extension part of the security system block which is described in Figure 2.1. The system consists of a camera node, a smart home server, a smart phone, an assistant robot and appliances in interacting with users in Figure 4.1.

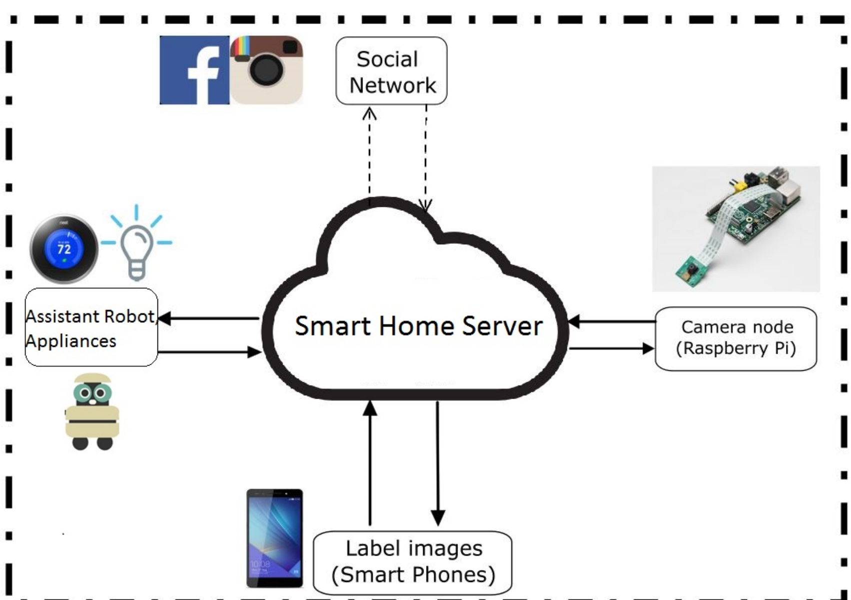

Figure 4.1: Home Security System Architecture.

Basically, the camera node will continuously run the face detection, whenever it detects a face, the image processing will be done locally on the camera node. It will match the face with the database and return the person's name with confidence and give a home access to user with level of access to smart home appliances. If the confidence is lower than the preset threshold, the camera will take 40 pictures and send them to the cloud. Then, the cloud will send a notification to the house's owner and ask for labeling the unidentified faces. After that, the new model will be trained in the smart home server and send back to the camera node to recognize the new faces.

4.1.1 Camera Node

The camera node uses a Raspberry Pi, a tiny and affordable computer, which is typically placed near the entrance where the access has to be granted. Whenever a person wants to access to the house, the camera node will capture an image, and process it further. The camera nodes are positioned so that it has a clear picture of the subject.

The camera node is in the waiting mode until it detects a person's face, and then it takes an image and runs the image processing locally in the Raspberry Pi by using Dlib [46] library and TensorFlow [7] which are installed inside the Raspberry Pi. The image is processed and compared with the datasets. The result returns the person's name and the confidence percentage.

4.1.2 Social Network

The social network node collects the data from the social networks such as Facebook, Instagram, etc. which are the largest free, diversified online data sources. There are two ways to collect the data from Facebook. First, we can create a public application from Facebook developer website. Second, we can collect photo data from an Android

App which runs Graph API [32]. These are easy ways to collect face images with labeled faces of users only if they give the access. We also mentioned before collecting data that the data will be used for research purpose and also is protected sensitive data for users.

4.2 Training and Incremental Learning

4.2.1 Introduction

Human brains make vision seem very easy, they do not take too much effort for us to tell apart a cheetah and a tiger, read a sign or recognize a human face. But these are really difficult problems to solve with a computer. In recent years, machine learning and neural network have made marvelous progress in solving these difficult problems. In particular, the model called a deep convolutional neural network can achieve nearly perfect performance on difficult visual recognition tasks. Researchers have worked on variant methods in computer vision by testing their works in ImageNet [47] - an academic benchmark for computer vision. Subsequent models improve each time to achieve a new record result: QuocNet [48], AlexNet [49], Inception (GoogLeNet), BN-Inception-v2 and Inception-v3 [50]. Inception-v3 is the latest trained model for the ImageNet Large Visual Recognition Challenge from Google. We implemented the face recognition module based on the method presented in Facenet [25] and the pre-trained Inception-v3 model in TensorFlow [7].

4.2.2 Face Recognition Architecture

The face recognition architecture is shown in Figure 4.2. The input data is aligned by using a face detection method, and then it goes through the deep convolutional neural network to extract embedding features. We can use the features for similarity detection and classification.

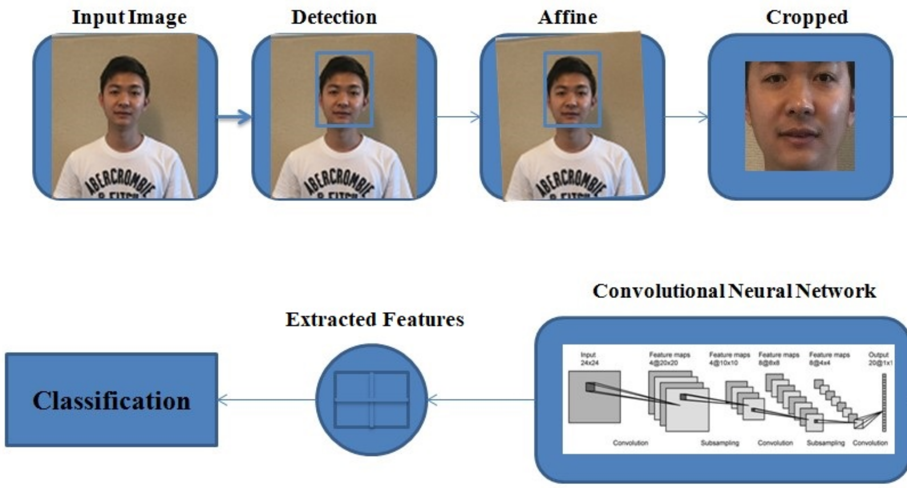

Figure 4.2: Face Recognition Architecture.

4.2.3 Detection and Affine Transformation

When processing an image, the face detection (Dlib library) [46] first finds a square around faces. Each face is then passed separately into the neural network, which expects a fixed size input, currently 96x96 pixels as mentioned in Facenet [25]. 96x96 pixels is the best size giving highest accuracy and low training time. We reshape the face in the square to 96x96 pixels. A potential issue with this is that faces could have different angles and we have to rotate the images. We use the align faces method described in OpenFace [51] by first finding the locations of the eyes and nose with Dlib's landmark detector. If the face is undetected or unaligned, the image will be eliminated before going to the neural network. Finally, an affine transformation is performed on the cropped image to make the eyes and nose appear at about the same place in all images as in Figure 4.3.

4.2.4 Initial Training Using Data from Open Sources

Our model trained 2622 celebrities from the VGG-Face dataset [52], 402 people from Facebook and 108 students using the security system. The process of collecting data of users and training the data is somewhat cumbersome. In order to make the training

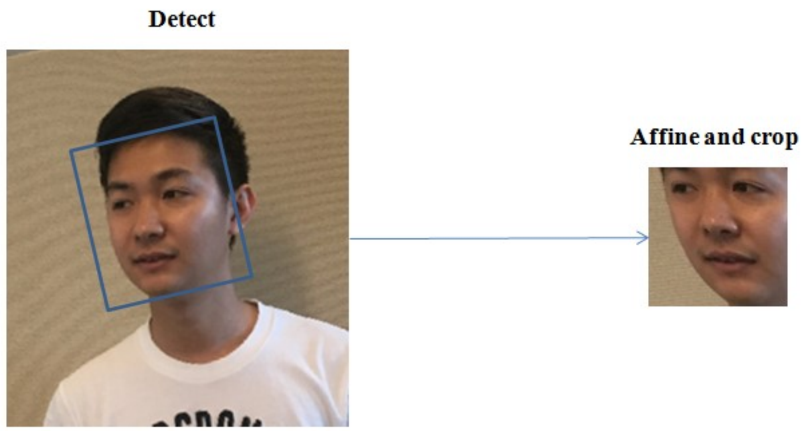

Figure 4.3: Cropped and Affined data.

process easier, we obtain the data from different social network accounts. Images of new users are obtained from the social media once the owner requests access for a particular user using the smart phone and new users after using the security system.

4.2.5 Incremental Learning

Often the training data obtained from the social network are insufficient for the deep learning model to perform accurately. Once the user is trained with a minimal dataset from the social media, the representation of the user is further improved by fine tuning the model as the user starts using the system. Sometimes the face recognition system fails to classify the person properly and will have a very low accuracy, in this case; the system asks help from the owner. A request is sent to the owner to label the unindented person through a smart phone. Afterward, the system automatically updates with the new data and sends back to the camera node to give the access. The interface is also built in a website and an android app which are friendly to label and collect data. We use the Triplet loss method mentioned in Facenet [25] for curriculum learning.

4.3 System Flowchart and Data Collection

4.3.1 System Flowchart

The system has two processing nodes. The first process is the camera node which is shown in Figure 4.4. By using a Raspberry Pi with a Pi camera, the camera node will detect and realize the human face with the current model and data stored in the Raspberry Pi memory. Then, it will give access or send data to the server based on whether the system is able to detect and recognize the face with a threshold confidence. If the confidence is lower than the threshold point, the Raspberry Pi will take a series of user images with different angles and expression to store on the server for later training. After the training task is done in the server, the updated model will be sent to the Raspberry Pi.

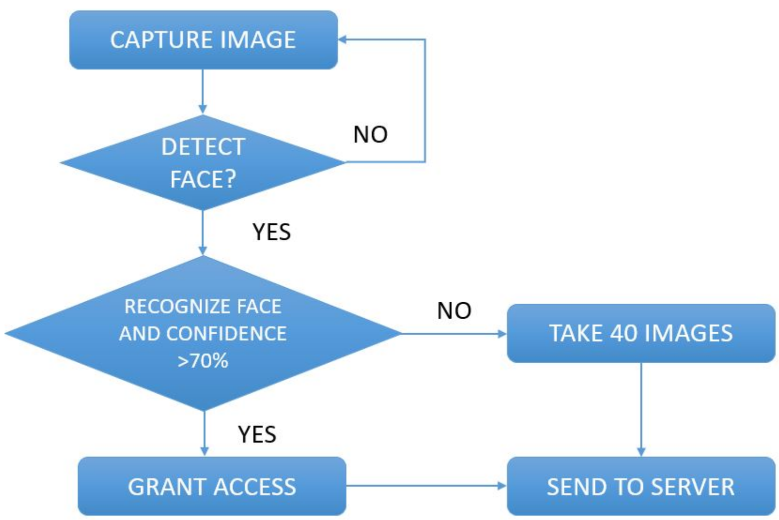

Figure 4.4: Camera Node Processes.

The second process is the server node which is shown in Figure 4.5. The server node aims to store the face data and send alerts to the owner asking for labeling the unidentified person. In addition, the Facebook web-based application is built in the server, which collects the data from social media. Afterward, we use the incremental learning technique in Facnet [25] to retrain the model with new data collected from

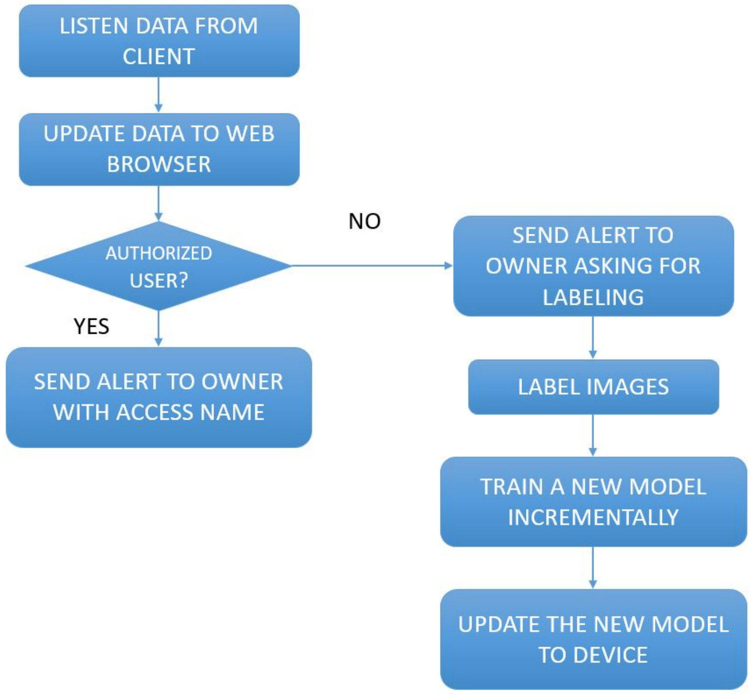

Figure 4.5: Server Node Processes.

social networks and the security systems.

4.3.2 Collecting Data from Facebook

The most important aspect of deep neural networks is data. As we mentioned in section 4.2.4 that the VGG dataset we found has only around 2.6 millions of images with around 2.6 thousand individuals. Compared with Google datasets mentioned in the Facenet paper [25], they use hundreds of millions of images from Google and Youtube. For the purpose of research or business, you have to pay for a robust face dataset or manually collecting the data will take a while, and the data are also insufficient. However, as social media becomes more popular around the world, we proposed a novel method to automatically collect the data from social media.

In this thesis, we only mention Facebook since it is the largest social network today, but actually we can use this method in other social media networks. By using the Graph API [32] for developers, we can extract the tag face from the users by giving the login access. We also built and published an application on Facebook which is very convenient for all users around the world who can log in and share their face images. The working flow of Facebook Graph API is shown in Figure 4.6.

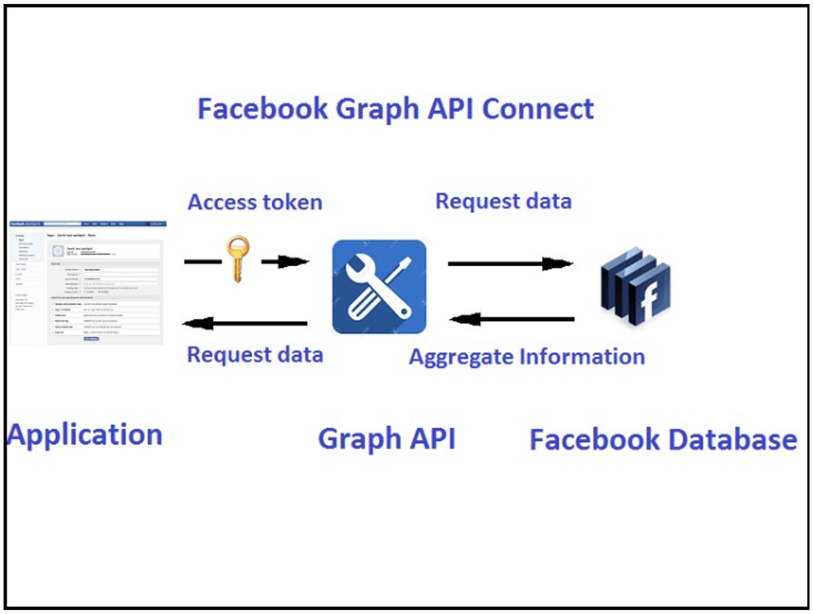

Figure 4.6: Facebook Graph API.

4.4 Experimental Results

4.4.1 Face Recognition Performance

First, we tested our model which was trained by the VGG dataset [52] on Labeled Faces in the Wild (LFW) datasets [27]. The classification accuracy is 0.974 ± 0.0140. The ROC of our model is shown in Figure 4.7 compared with Human and Eigenfaces experiments. Unfortunately, the model is unable to reach the accuracy mentioned in the Facenet paper [25] since we use less input data than the paper. However, the accuracy is obviously impressive to compare with Eigenfaces algorithm used in Jeffrey's [53] and Shankar's [54] security system. The state-of-art Inception-v3 model gives a marvelous result.

We also tested the face recognition system in reality by using 20 subjects as shown

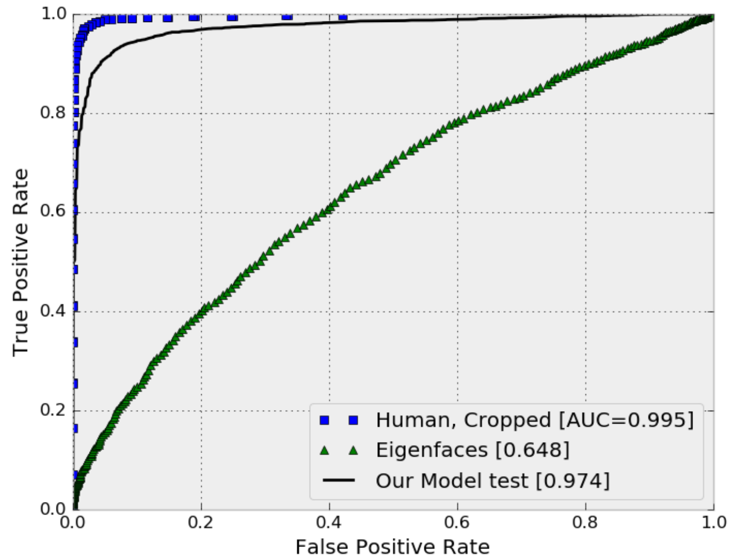

Figure 4.7: Our Model Test ROC with LFW.

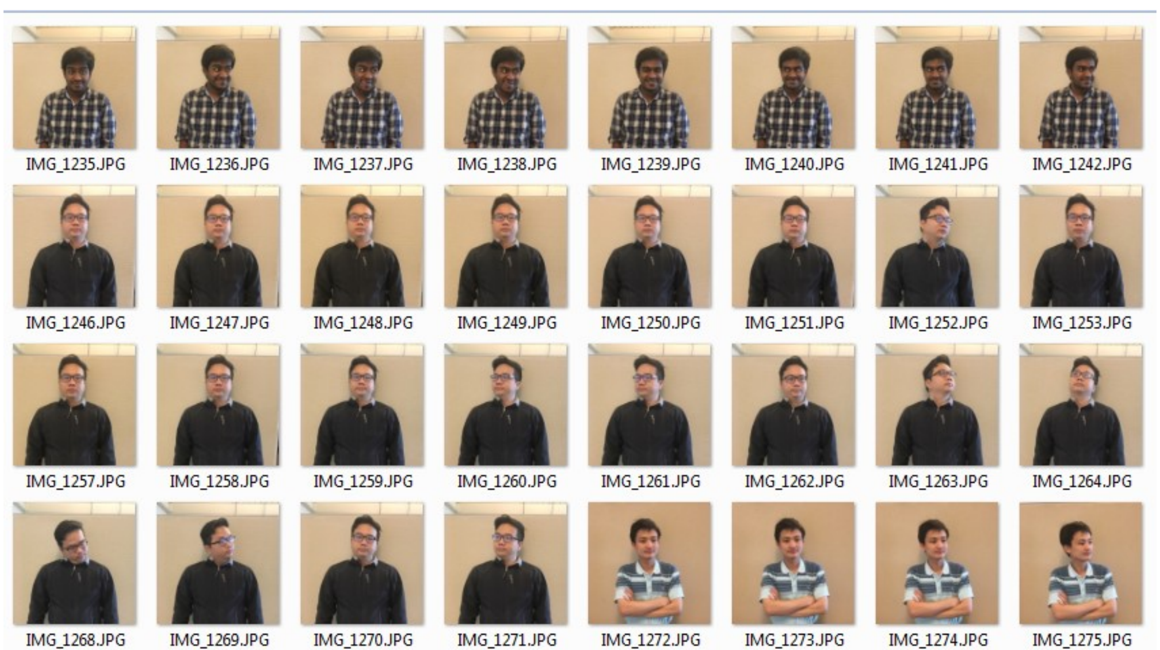

Figure 4.8: Lab Members Dataset.

in Figure 4.8. The highest accuracy is 92.2%.

These people are presented as new guests coming to the house. Since the dataset we used to train the neural network model is all American while half of the test data is Asian, the result has a low accuracy and sometimes it failed to distinguish people. The system sometimes gets confused between two people with similar faces, but more face images with different angles and expressions solved the problem. The light condition is also important and the background should not be too illuminated. The bottleneck values are shown in Figure 4.9 before training and those values will be grouped to different class after training.

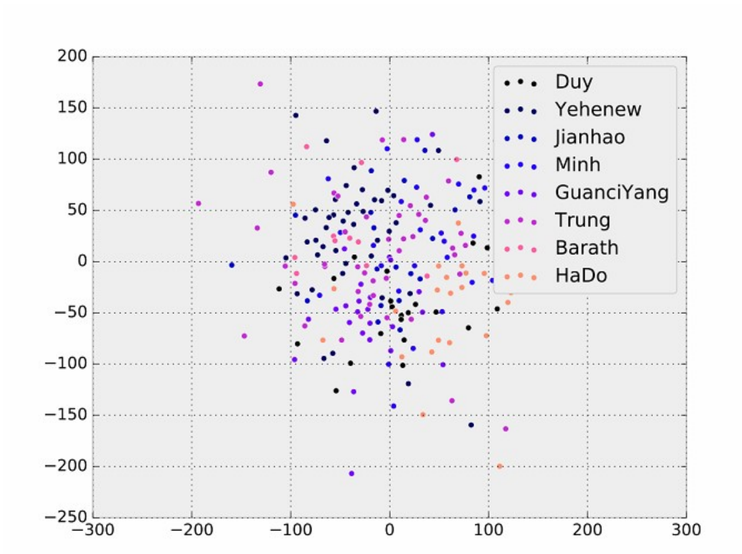

Figure 4.9: Bottleneck Values.

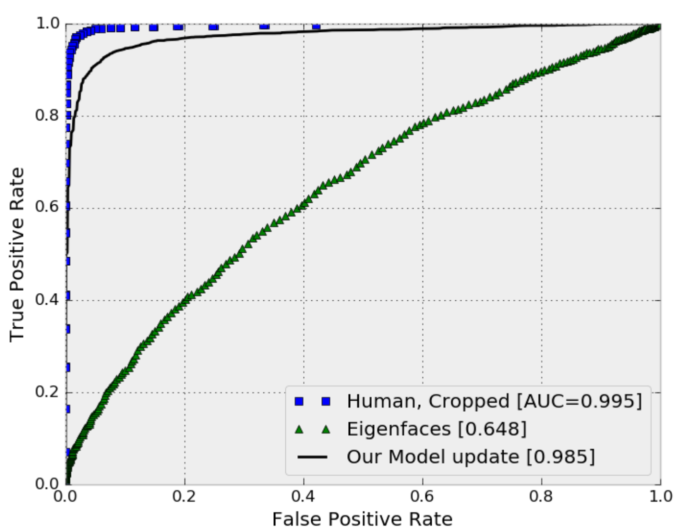

Figure 4.10: Our Model Update ROC.

After collecting the data from 108 students and 402 users from Facebook, we trained a new model with the updated data using incremental learning. The result is shown in Figure 4.10 with improving AUC from 0.974 to 0.985.

Also, the number of correctly recognized faces increased because we updated the data with diverse images of people from different regions. We would not expect the accuracy will dramatically increase because the data we collected is insufficient and also at the limit of the algorithm.

4.4.2 Security System Setup

Figure 4.11: Camera Node (Raspberry Pi).

The entire system was developed and tested in a miniature smart home which imitates the actual smart home. The miniature smart home with the Raspberry Pi camera node is shown in Figure 4.11.

4.4.3 Interface Collecting Data from Social Media and Human

We developed an android app to alert the owner/administrator via a smart phone which is shown in Figure 4.12. The left side is the interface with the owner, whenever someone tries to access the house, the new image will be updated in the app and on the website as well. The owner will receive a notification, which is labeled by the name or identified as an unknown face. Then the owner can label new users and the system will automatically retrain the classifier model with new users and give them access. On the right side, the Facebook graph API is built into the app with the necessary information from Facebook's database. We collected the tagged faces with face locations and saved to the server storage node. We also gave a permission level for different users which will protect privacy or children as shown in Figure 4.13. For example, the guests do not have access to control the bedroom door, and the kids cannot control the dangerous electric devices or television.

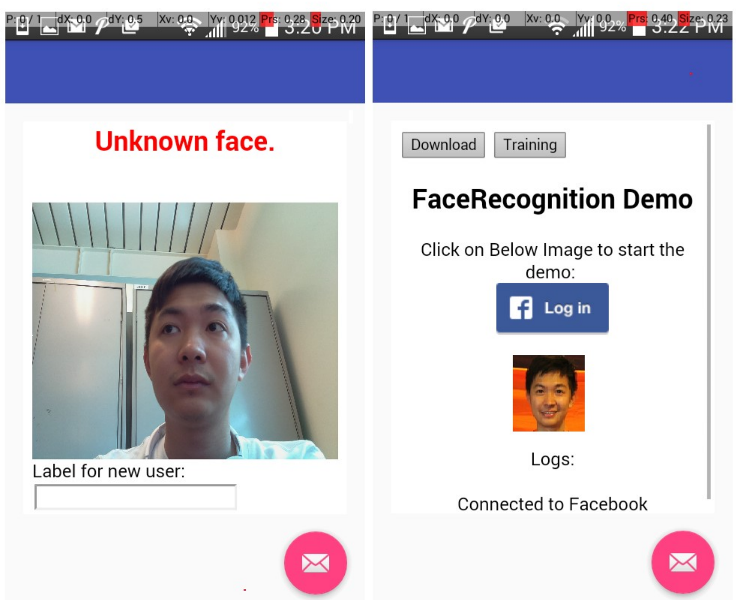

Figure 4.12: Android Application: Users Interface (left) and Facebook Interface.

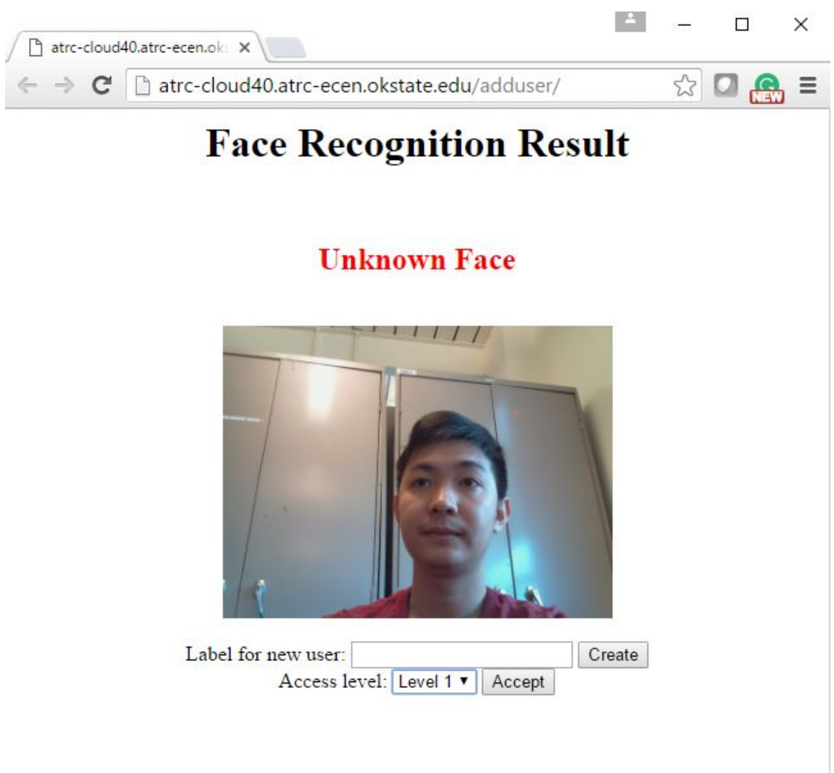

Figure 4.13: Web Browser Interface.

CHAPTER 5

SMART HOME ASSISTANT ROBOT

In this chapter, we describe our development of the smart home assistant robot which includes voice controller, Wechat controller, and action and event recognition features.

5.1 Smart Home Assistant

The smart home assistant is an assistant robot which can control the smart home devices through a person's voice and a social media application by connecting to public clouds such as Google, and Wechat. The robot contains a camera which can be used to run the security system features.

5.1.1 Voice Controller

The voice controller, based on Mycroft [11], acts as a chatting robot in the smart home. The Mycroft uses Google API to convert speech to text and then locally processes the text through natural language understanding before acting on the smart home or responding to the user. The process is shown in Figure 5.1. Google API and Mycroft provide accurate and reliable voice recognition and natural language processing capabilities. The voice controller includes the following four steps:

Setup one:

Mycroft module runs locally on the BI robot, a smart home assistant robot developed in ASCC lab. A person's voice first enters the BI robot through a microphone; A voice command set is programmed on the Mycroft module which helps users to simply

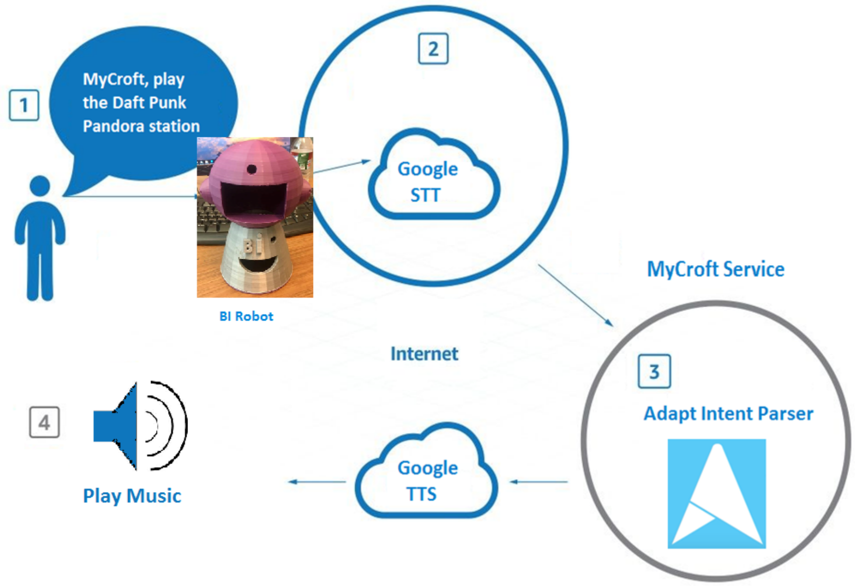

Figure 5.1: Voice Controller System.

add new skills. We define the keywords to enable the related skills and define the responding sentences from BI robot.

Step two:

The recorded audio will be sent to Google API to get converted from speech to text. The speech-to-text API is a flexible choice since many big companies support this feature such as IBM Watson [55], Amazon Alexa [9], PocketSphinx [56], etc. However, Google is leading in this field and it APIs are easier to use compared with other competitors.

Step three:

After texts were extracted from Google API, it goes through a natural language processing step to understand the content of the texts. An Adapt Intent Parser [57] is installed in the BI robot to do this job. The Adapt Intent Parser is a lightweight parser and is designed to run on devices with limited computing resources.

Step four:

This step includes voice command skill sets which are programmed on the Mycroft module to control the smart home appliances. The skill sets consist of turning on and off lights, setting up the temperature, and changing air conditioner modes. The BI robot responds to users by using Google Text-to-Speech API, Mimic [58], or Espeak [59].

5.1.2 Wechat Controller

The Wechat controller system which is developed by Chengjie Lin is shown in Figure 5.2 which helps users control and monitor the smart home testbed remotely through conversation. This system is developed based on Wechat [37], a social network software similar to Whatsapp [60]. The system consists of the iFlytek voice processing cloud [61], Wechat website cloud, and Turing intelligent conversation cloud [62]. It allows the users to control the smart home via Wechat app easily.

The Wechat controller system is currently targeted at Chinese-speaking users, however, the smart home controlling and monitoring features also work in other languages. The current users of Wechat are around 600 million, and it is convenient for users to update to get new features for smart home appliances. The current functions of this Wechat controller system include setting the temperature, checking the temperature, setting the AC mode, controlling and checking the light. For example, if a user sends a request to the Wechat app to turn off the light by voice or text messages, the text or voice commands will first be sent to the Wechat website cloud, and then encoded and forwarded to the smart home assistant. Then, those voice commands will be resampled to the correct format to optimize the speech to text conversion process. The modified voice file will be uploaded to the iFlytek voice processing cloud [61] and converted to the text, and the text will be sent to the smart home assistant which uploads the text to the Turing intelligent conversation cloud [62] to utilize the

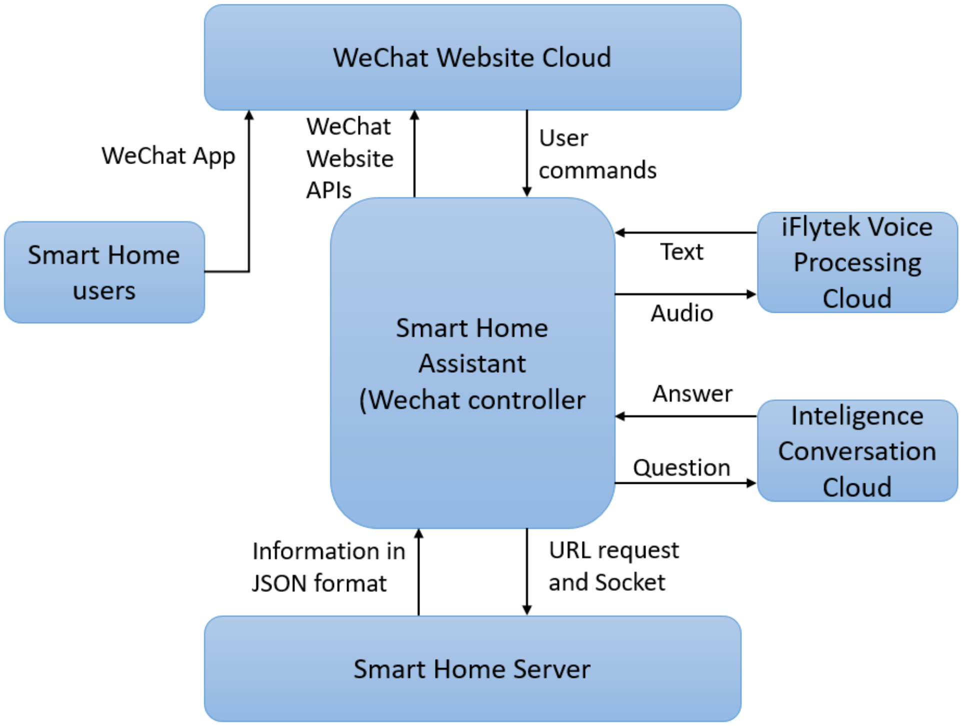

Figure 5.2: Wechat Controller System.

Natural Language Processing. Next, the text is parsed to the JSON format to get the keyword. It will call the corresponding smart home control APIs that are indicated by the keyword.

The controlling part of Smart Home APIs is based on sending URL requests with the JSON-formatted commands to the smart home server. Whenever a text message is sent to the Wechat cloud server, it will give a command to control the smart home devices or query the smart home status.

5.2 Action and Event Recognition (Show and Tell)

By including a Raspberry Pi camera in the home assistant robot, we are able to run the face recognition feature which is described in Chapter 4. Also, taking the advantage of the convolutional neural network (CNN), we combine with a long short-term memory (LSTM) network to develop an action and event recognition system

(Show and Tell).

The Show and Tell model is an example of an encoder-decoder neural network. First, the model will 'encode' an image into a fixed-length vector representation, and then 'decode' the representation into a natural language description. The image encoder is a deep convolutional neural network. The name of the network is the Inception v3 image recognition model [7] pre-trained on the ImageNet image classification dataset [47]. The decoder is a long short-term memory (LSTM) network. This network is commonly used for sequence modeling tasks such as language modeling. Briefly, the model will first detect and recognize the object and action inside the image and convert it to text representation, from texts, the model will generate a make-sense sentence to describe the image. The action and event recognition system is shown in Figure 5.3. From a security camera or an assistant robot, pictures will be taken and sent to the smart home server, a show and tell model which combines a CNN network and LSTM network will convert images to contents. From text contents, we use an early warning detection system which detects dangerous or warning keywords such as gun, knife, fire, smoke, falling, fighting, etc. and send back to the assistant robot to notice the users.

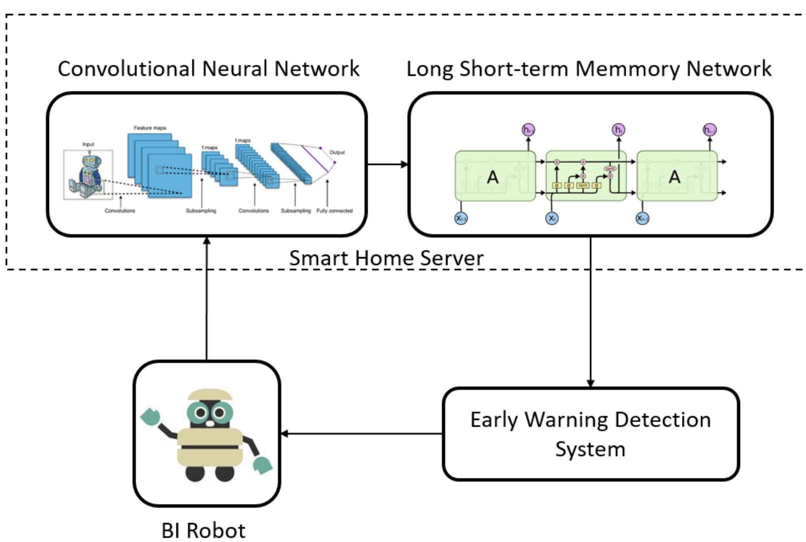

Figure 5.3: Action and Event Recognition System.

5.3 Experimental Results

5.3.1 Voice Controller

After implementing Mycroft on Raspberry Pi, we tested with a set of questions, then recorded the responses. We also created Mycroft skills to control the smart home. We tried asking questions from various sectors and in different formats and have received accurate and reliable results. Table 5.1 shows the asked questions and the received responses.

No.	Queries	Response
1	BI Robot, wake up	"Hello, I'am BI Robot"
2	BI Robot, how is the weather today	"In Stillwater, It's 44 degree, clear sky..."
3	BI Robot, how about tomorrow	"Tomorrow, in Stillwater, 45 degree... "
4	BI Robot, how about in Dallas	"Right now in Dallas, it's 52 degree..."
5	BI Robot, tell me a joke	Who is not hungry at Thanksgiving? The turkey because he's already stuffed!
6	BI Robot, sing Happy Birthday song	"Happy birthday to you ..."
7	BI Robot, wake me up at 9 in the morning	"Alarm is set for 9am tomorrow"
8	BI Robot, what is the capital of Oklahoma	"Capital of Oklahoma is Oklahoma City"
9	BI Robot, set a reminder to submit my assignment by the end of the day	Submit my assigment is add to my to-do list
10	BI Robot, what is on my to-do list	You have one item in the to-do list: your assignment by the end of the day
11	BI Robot, what is in the news	"From NPR news in Washington DC ...
12	BI Robot, stop	"Yes"
13	BI Robot, Turn on the kitchen light	"Kitchen light is on"
14	BI Robot, Turn on the kitchen light	"Kitchen light is already on"
15	BI Robot, Turn off the kitchen light	"Kitchen light is off"

Table 5.1: Voice Controller Test Scenario.

5.3.2 Show and Tell Model

We took several scenarios which are activities at home to test the Show and Tell model. The result is very impressive and is shown in Figure 5.4.

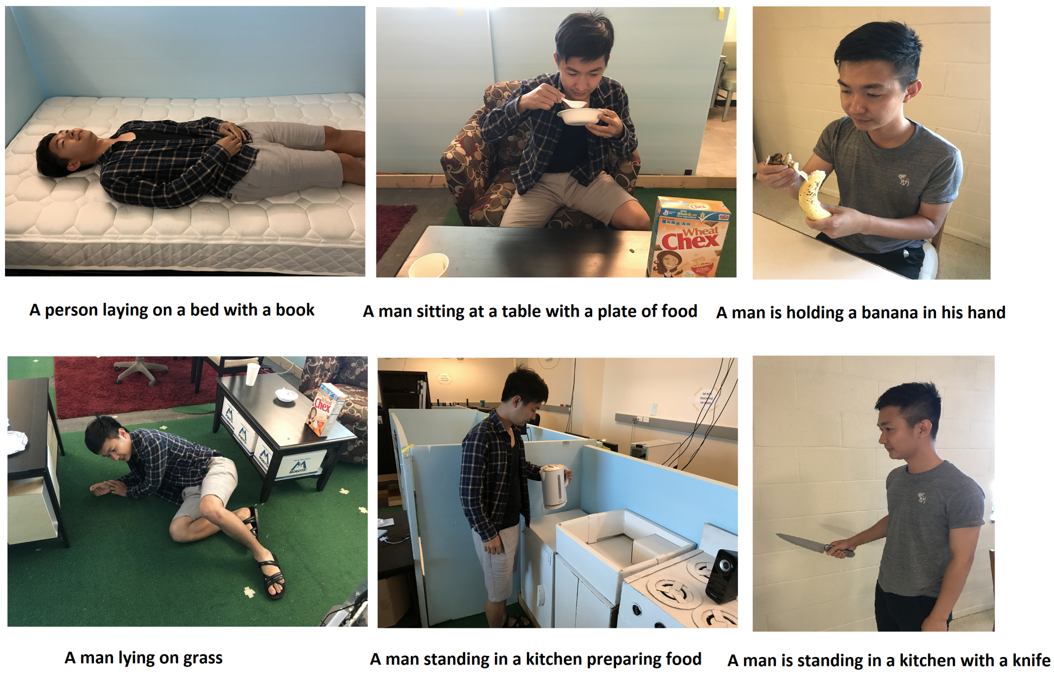

Figure 5.4: Show and Tell Test Scenario

CHAPTER 6

CONCLUSIONS

6.1 Discussions

This thesis introduced a mini smart home testbed along with a hardware/software framework which can help organize, configure and control the smart home remotely from any clients and anywhere. It also proposed an energy saving method that considers both the power consumption and generation. The home controller gathers sensor values and sends to the smart home server. The smart home server manages the energy conservation method to save the energy. The paper also described a security system with face recognition and early warning detection. We also developed a smart home assistant robot which can control the smart home appliances from the person's voice and the Wechat app. This architecture and framework created a ubiquitous platform to support education and research in various areas such as IoT, artificial intelligence, renewable energy, etc.

The theme of this thesis is about smart home systems. The following sections are what we have learned after implemented the system on the miniature smart home testbed.

6.1.1 Smart Home System

The most advanced part of our smart home setup is the smart home server. The previous works on smart home systems use local solution. By using a cloud service for smart home server, we no longer need a powerful machine for the home controller, and other features like speech-to-text, text-to-speech, deep neural network work the best on cloud services.

6.1.2 Smart Energy Management System

The most important part of the smart energy management system is the communication. How we can monitor all smart devices and how to control those devices the fastest way. Since we have all information from appliances, how to use that information to save more power is based on testing and researching about human behaviors and smart home environments.

6.1.3 Face Recognition for Security Systems

As we keep repeat it again and again, the most important thing for face recognition or deep neural network is data. How we collect the data, how we pre-process the data effect to the final model. We learned a new way to collect the data from social medias but how to pre-process those data still is a challenge.

Our face recognition model reached to 0.985 % in testing datasets. However, we are still able to increase the performance. The face crop and affine still need to improve by removing the background, extracting only the face out-shape, and affining all faces.

6.1.4 Smart Home Assistant Robot

The smart home robot is the future of smart home systems. We built a platform for the smart home assistant robot (BI Robot). The challenge is how to develop applications for this robot because without the skills and applications, the robot is useless.

6.2 Future Works

In the future, we will focus on building a smart home system with more applications such as smart sensor network systems, smart health center systems, etc. By adopting machine learning algorithms, we can develop a genuinely smart home. Moreover, the gesture-controlled feature [63] and elderly care services [64] will be integrated into the BI robot.

To improve the system, we propose some technologies which are enhanced the smart home system to a different level and able to apply to real smart home systems.

6.2.1 Energy Saving through Human Activity Awareness

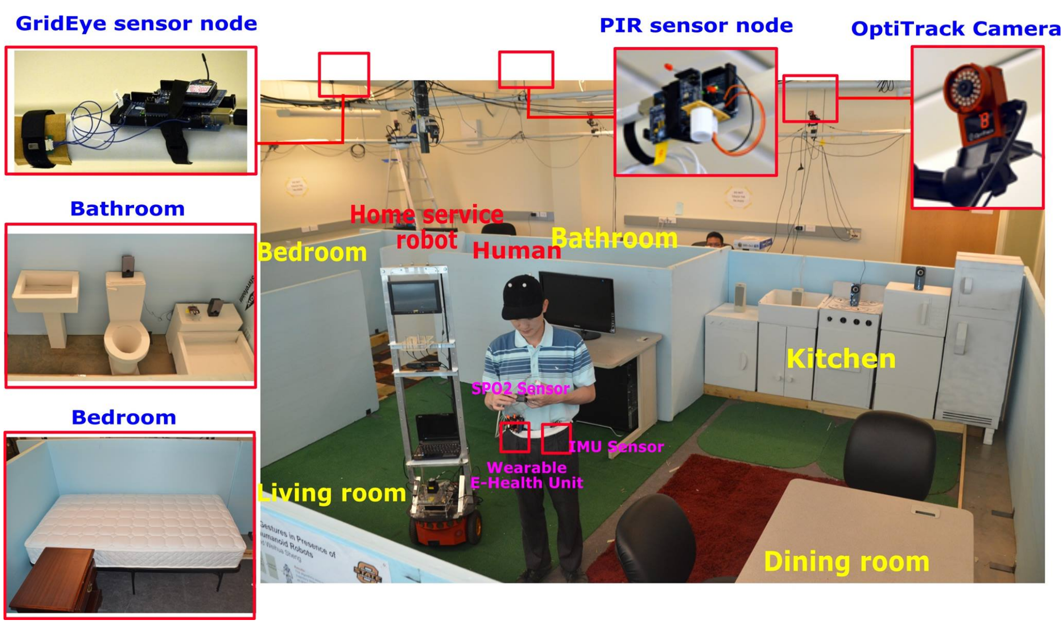

Figure 6.1: Smart Home PIR Sensors and Wearable Devices.

By matching human activities through smart home sensors and wearable smart devices to the smart home system, we are able to save more energy and make home smarter. The ideal scenario is when the home sensors detect human inside the house and which room people are not in, the system will try to dim the light and then turn it off after 30 minutes. Also, based on the human activities, when people are sleeping or exercising, the AC system will adjust the temperature to save the energy. However, we only implemented the human detection part using PIR and leave other to the future works. The smart home PIR sensors and wearable devices are shown in Figure 6.1 which is developed by Minh Pham and HaDo [65] [66].

6.2.2 Wireless Sensor Network

The miniature smart home testbed is designed to use the GPIO Raspberry Pi to connect to appliances, but we also developed a wireless connection between the home controller and actuators. The wireless module is shown in Figure 6.2 (right) with MQTT, Wifi protocol. In the current setup, the wireless module connects to an Amazon server and change some parameter on the server, but in the future, we will implement this feature on our smart home system with more support protocol such as ZigBee and Z-Wave.

The home controller and smart home server installed all necessary protocols such as ZigBee, Z-wave, and MQTT. The ideal goal is that each device is low-power consumed, low cost, connect through ZigBee or Z-wave. Then, the home controller will update all data to the smart home server through MQTT communication as shown in 6.2 (left).

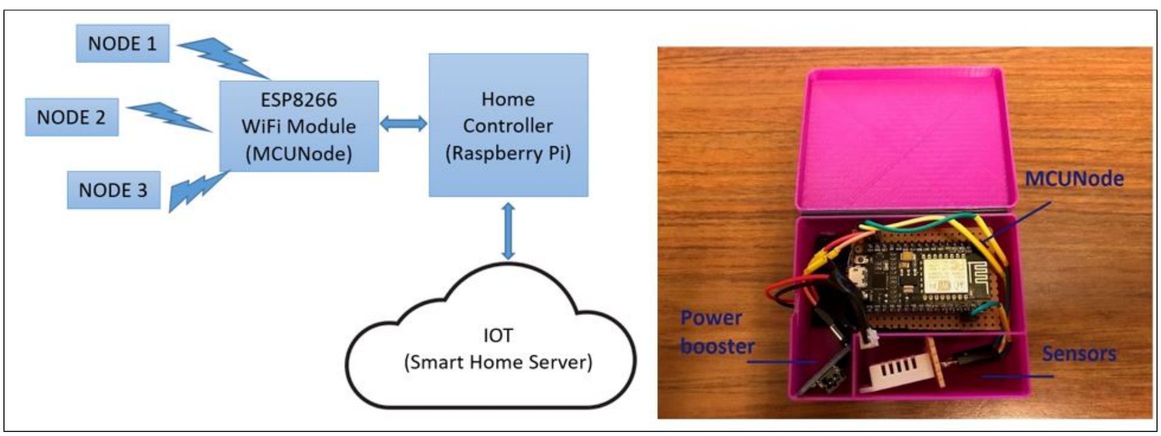

Figure 6.2: Wireless Sensor Network (left) and Wireless Module (right)

6.2.3 BI Robot Design

We encourage to build your own AI Assistant Robot. BI Robot will become a household phenomenon, more advanced than current assistant robots in the market with the main features: interfacing with robot's face, supporting a camera, answering questions, playing music, controlling smart devices, recognizing faces and detecting warning situation. The design is shown in Figure 6.3. The current design still has some limits and need to redesign.

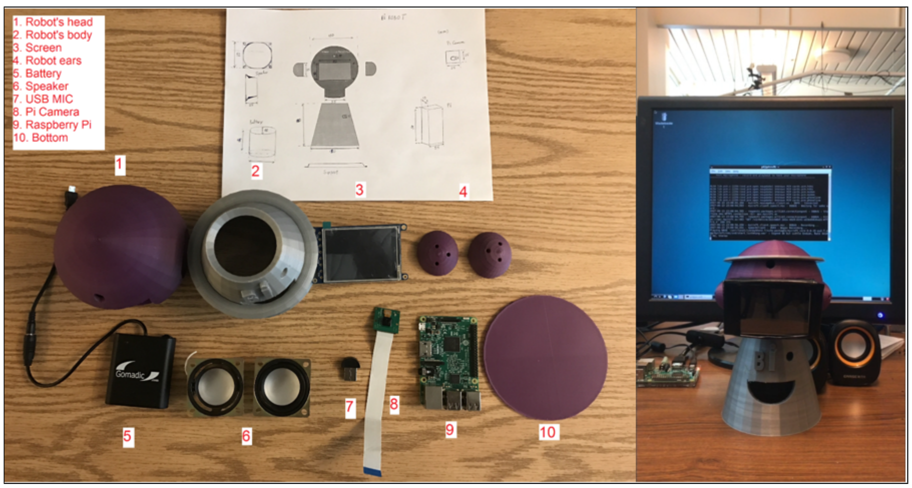

Figure 6.3: BI Robot Design

6.2.4 Show and Tell Model

Since we only use the show and tell model in smart homes. We should collect and train a dataset which is only including home activities and objects to make the model lighter, faster, and more accurate.

6.2.5 Collecting Data for Face Recognition

Since we proposed few methods to collect the data from a smart phone application and social medias but we have not really focused on making that running smoothly as the flow chart but testing. However, we proposed the complete picture for the system which you will have to finish some bellow tasks to fill the big picture:

- Adding notifications to the owner when unidentified person presenting in the front door;
- Asking the unidentified person to change different angles when collecting images;
- Automatically download the new training model to the camera node and continue to recognize faces.

6.2.6 Power Consumption

The current setup used some fixed values from appliances but in order to have more reasonable and accurate scenario, we should use current sensors in all smart devices to track the real time consumed power. When the power consumptions are correct, the system will run as the method we proposed.

BIBLIOGRAPHY

[1] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Review of communication technologies for smart homes/building applications," in *Innovative Smart Grid Technologies-Asia (ISGT ASIA), 2015 IEEE*, pp. 1–6, IEEE, 2015.

[2] H. M. Do, W. Sheng, M. Liu, and S. Zhang, "Context-aware sound event recognition for home service robots," in *Automation Science and Engineering (CASE), 2016 IEEE International Conference on*, pp. 739–744, IEEE, 2016.

[3] H. M. Do, C. J. Mouser, Y. Gu, W. Sheng, S. Honarvar, and T. Chen, "An open platform telepresence robot with natural human interface," in *Cyber Technology in Automation, Control and Intelligent Systems (CYBER), 2013 IEEE 3rd Annual International Conference on*, pp. 81–86, IEEE, 2013.

[4] H. M. Do, W. Sheng, and M. Liu, "An open platform of auditory perception for home service robots," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 6161–6166, IEEE, 2015.

[5] P. P. Gaikwad, J. P. Gabhane, and S. S. Golait, "A survey based on smart homes system using internet-of-things," in *Computation of Power, Energy Information and Commuincation (ICCPEIC), 2015 International Conference on*, pp. 0330–0335, IEEE, 2015.

[6] T. Nguyen, L. Barth, W. Sheng, Y. Gu, M. Liu, and S. Zhang, "A miniature smart home testbed for research and education," in *Cyber Technology in Automation, Control and Intelligent Systems (CYBER), 2017 IEEE 7th Annual International Conference on*, pp. 81–86, IEEE, 2017.

[7] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[8] P. Kim, "Convolutional neural network," in *MATLAB Deep Learning*, pp. 121–147, Springer, 2017.

[9] F. Azam, A. Shafiq, and Aneeq-Ur-Rehman, "Smart home system," tech. rep., COMSATS Institute of Information Technology Lahore-Pakistan, 2015.

[10] A. Nijholt, "Google home: Experience, support and re-experience of social home activities," *Information Sciences*, vol. 178, no. 3, pp. 612–630, 2008.

[11] S. Penrod, "Mycroft, an open source artificial intelligence for everyone." `https://github.com/MycroftAI`, 2017.

[12] J. P. Mueller, *Mining Google? Web Services: Building Applications with the Google? API.* John Wiley & Sons, 2006.

[13] D. M. Han and J. H. Lim, "Design and implementation of smart home energy management systems based on zigbee," *IEEE Transactions on Consumer Electronics*, vol. 56, pp. 1417–1425, Aug 2010.

[14] D.-M. Han and J.-H. Lim, "Smart home energy management system using ieee 802.15. 4 and zigbee," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, 2010.

[15] S. Kumar, "Ubiquitous smart home system using android application," *arXiv preprint arXiv:1402.2114*, 2014.

[16] M. A. A. Pedrasa, T. D. Spooner, and I. F. MacGill, "Coordinated scheduling of residential distributed energy resources to optimize smart home energy services," *IEEE Transactions on Smart Grid*, vol. 1, no. 2, pp. 134–143, 2010.

[17] Q. Hu and F. Li, "Hardware design of smart home energy management system with dynamic price response," *IEEE Transactions on Smart grid*, vol. 4, no. 4, pp. 1878–1887, 2013.

[18] M. Jahn, M. Jentsch, C. R. Prause, F. Pramudianto, A. Al-Akkad, and R. Reiners, "The energy aware smart home," in *Future Information Technology (FutureTech), 2010 5th International Conference on*, pp. 1–8, IEEE, 2010.

[19] ADT, "Security systems, home automation, alarms and surveillance." `http://http://www.adt.com/`, 2016. [Accessed: 2016-09-09].

[20] V. S. H. S. Systems, "Smart home security solutions.." `http://http://www.vivint.com/`, 2016. [Accessed: 2016-09-09].

[21] P. America, "Affordable home securitysystems for everyone.." `http://www.protectamerica.com/`, 2016. [Accessed: 2016-09-09].

[22] K. Denmead, "Netatmo is the weather station for the rest of us. wired magazine.[online]. http," *archive. wired. com/geekdad/2013/02/netatmo/.[Accessed 2014-04-02]*, 2013.

[23] H. M. Do, C. Mouser, M. Liu, and W. Sheng, "Human-robot collaboration in a mobile visual sensor network," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 2203–2208, IEEE, 2014.

[24] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[25] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

[27] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," tech. rep., Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[28] I. Kuzborskij, F. Orabona, and B. Caputo, "From n to n+ 1: Multiclass transfer incremental learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3358–3365, 2013.

[29] H. M. Do, W. Sheng, and M. Liu, "Human-assisted sound event recognition for home service robots," *Robotics and Biomimetics*, vol. 3, p. 7, Jun 2016.

[30] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, ACM, 2009.

[31] T. Xiao, J. Zhang, K. Yang, Y. Peng, and Z. Zhang, "Error-driven incremental learning in deep convolutional neural network for large-scale image classification," in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 177–186, ACM, 2014.

[32] Facebook, "Graph api, facebook for developers." `https://developers.facebook.com/docs/graph-api`, 2017. [Accessed: 2017-06-24].

[33] A. Colon and M. Greenwald, "Amazon echo," *PC Magazine Internet page located at: http://www. pcmag. com/article2/0*, vol. 2817, no. 2476678, p. 00, 2015.

[34] J. Zhu, X. Gao, Y. Yang, H. Li, Z. Ai, and X. Cui, "Developing a voice control system for zigbee-based home automation networks," in *Network Infrastructure and Digital Content, 2010 2nd IEEE International Conference on*, pp. 737–741, IEEE, 2010.

[35] J.-K. Guo, C.-L. Lu, J.-Y. Chang, Y.-J. Li, Y.-C. Huang, F.-J. Lu, and C.-W. Hsu, "Interactive voice-controller applied to home automation," in *Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP'09. Fifth International Conference on*, pp. 828–831, IEEE, 2009.

[36] M. Sripan, X. Lin, P. Petchlorlean, and M. Ketcham, "Research and thinking of smart home technology," in *International Conference on Systems and Electronic Engineering-(ICSEE'2012*, 2012.

[37] WeChat, "The new way to connect." `https://www.wechat.com`, 2017. [Accessed: 2017-05-15].

[38] K. Banker, *MongoDB in action*. Manning Publications Co., 2011.

[39] M. Masse, *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. " O'Reilly Media, Inc.", 2011.

[40] H. Assistant, "An open source home automation platform." `https://home-assistant.io/`, 2017. [Accessed: 2017-06-24].

[41] NEST, "Nest thermostat." `https://nest.com/`, 2017. [Accessed: 2017-06-24].

[42] A. Banks and R. Gupta, "Mqtt version 3.1. 1," *OASIS standard*, vol. 29, 2014.

[43] Z. Alliance, "Ieee 802.15. 4, zigbee standard," 2009.

[44] M. T. Galeev, "Catching the z-wave," *Embedded Systems Design*, vol. 19, no. 10, p. 28, 2006.

[45] F. Bellard, "Qemu-open source processor emulator." `http://wiki.qemu.org/Main_Page`, 2016. [Accessed: 2016-09-09].

[46] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1755–1758, 2009.

[47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255, IEEE, 2009.

[48] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 8595–8598, IEEE, 2013.

[49] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[50] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *arXiv preprint arXiv:1512.00567*, 2015.

[51] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," tech. rep., Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

[52] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*, 2015.

[53] J. S. Coffin and D. Ingram, "Facial recognition system for security access and identification," Nov. 23 1999. US Patent 5,991,429.

[54] J. S. Kartik, K. R. Kumar, and V. Srimadhavan, "Security system with face recognition, sms alert and embedded network video monitoring terminal," *International Journal of Security, Privacy and Trust Management (IJSPTM) Vol*, vol. 2, 2013.

[55] I. Watson, "Speech to text, watson developer cloud." `https://www.ibm.com/watson/developercloud/doc/speech-to-text/index.html`, 2017. [Accessed: 2017-06-24].

[56] D. Huggins-Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar, and A. I. Rudnicky, "Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1, pp. I–I, IEEE, 2006.

[57] Mycroft, "Adapt intent parser, an open source natural language understanding." `https://adapt.mycroft.ai`, 2017. [Accessed: 2017-06-24].

[58] Mycroft, "Mimic, an open source text to speech." `https://mimic.mycroft.ai`, 2017. [Accessed: 2017-06-24].

[59] Espeak, "Espeak, an open source text to speech." `http://espeak.sourceforge.net/`, 2017. [Accessed: 2017-06-24].

[60] WhatsApp, "Whatsapp.com." `https://www.whatsapp.com`, 2017. [Accessed: 2017-05-15].

[61] iFlytek, "Speech recognition." `http://www.iflytek.com`, 2017. [Accessed: 2017-05-15].

[62] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, no. 236, pp. 433–460, 1950.

[63] Y. Gu, H. Do, Y. Ou, and W. Sheng, "Human gesture recognition through a kinect sensor," in *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, pp. 1379–1384, IEEE, 2012.

[64] Y. Mengistu, M. Pham, H. M. Do, and W. Sheng, "Autohydrate: A wearable hydration monitoring system," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pp. 1857–1862, IEEE, 2016.

[65] F. E. Fernandes, G. Yang, H. M. Do, and W. Sheng, "Detection of privacy-sensitive situations for social robots in smart homes," in *Automation Science and Engineering (CASE), 2016 IEEE International Conference on*, pp. 727–732, IEEE, 2016.

[66] M. Pham, Y. Mengistu, H. M. Do, and W. Sheng, "Cloud-based smart home environment (coshe) for home healthcare," in *Automation Science and Engineering (CASE), 2016 IEEE International Conference on*, pp. 483–488, IEEE, 2016.

[67] L. Lamport, *LaTeX User's Guide and Reference Manual*. Addison Wesley, 2nd ed., 1994.

[68] N. Skeledzija, J. Cesic, E. Koco, V. Bachler, H. N. Vucemilo, and H. Dzapo, "Smart home automation system for energy efficient housing," in *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*, pp. 166–171, IEEE, 2014.

[69] Y.-G. Ha, "Dynamic integration of zigbee home networks into home gateways using osgi service registry," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, 2009.

[70] S. Hickson, "Steven hickson voicecommand." `https://github.com/StevenHickson/PiAUISuite`, 2017. [Accessed: 2017-03-03].

[71] F. B. o. I. United States Department of Justice, "Crime in the united states, 2015." https://ucr.fbi.gov/crime-in-the-u.s/2014/crime-in-the-u.s.-2014/resource-pages/about-cius/, 2015.

[72] P. I. Wilson and J. Fernandez, "Facial feature detection using haar classifiers," *Journal of Computing Sciences in Colleges*, vol. 21, no. 4, pp. 127–133, 2006.

[73] R. J. Qian, M. I. Sezan, and K. E. Matthews, "A robust real-time face tracking algorithm," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, vol. 1, pp. 131–135, IEEE, 1998.

[74] Soumith, "Convnet-benchmarks." https://github.com/soumith/convnet-benchmarks, 2016. [Accessed: 2016-09-09].

[75] H. M. Do, C. J. Mouser, and W. Sheng, "Building a telepresence robot based on an open-source robot operating system and android," in *Third Conference on Theoretical and Applied Computer Science (TACS 2012)*, 2012.

[76] H. M. Do, "Developing a home service robot platform for smart homes," Master's thesis, Oklahoma State University, 2015.

VITA

Trung Nguyen

Candidate for the Degree of

Master of Science

Dissertation: DEVELOPMENT OF A MINIATURE SMART HOME TESTBED FOR RESEARCH AND EDUCATION.

Major Field: Electrical and Computer Engineering

Biographical:

Personal Data: Born in Thainguyen, Vietnam on 11 25, 1991.

Education:

Received the B.S. degree from Posts and Telecommunications Institute of Technology, Hanoi, Vietnam, 2017, in Electronics and Telecommunications Engineering Major

Experience:

A dedicated and motivated researcher with 2+ years experience in machine learning and computer vision with strong coding skills, solid background in hardware and mathematics. Specializes in using deep neural networks in pattern recognition and natural language processing on cloud computing and embedded computer systems.