Applied Artificial Intelligence
An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Smart Control of Home Appliances Using Hand Gesture Recognition in an IoT-Enabled System

Cheng-Ying Yang, Yi-Nan Lin, Sheng-Kuan Wang, Victor R.L. Shen, Yi-Chih Tung, Frank H.C. Shen & Chun-Hsiang Huang

To cite this article: Cheng-Ying Yang, Yi-Nan Lin, Sheng-Kuan Wang, Victor R.L. Shen, Yi-Chih Tung, Frank H.C. Shen & Chun-Hsiang Huang (2023) Smart Control of Home Appliances Using Hand Gesture Recognition in an IoT-Enabled System, Applied Artificial Intelligence, 37:1, 2176607, DOI: 10.1080/08839514.2023.2176607

To link to this article: https://doi.org/10.1080/08839514.2023.2176607

Published online: 15 Feb 2023.

APPLIED ARTIFICIAL INTELLIGENCE
2023, VOL. 37, NO. 1, e2176607 (647 pages)
https://doi.org/10.1080/08839514.2023.2176607

Smart Control of Home Appliances Using Hand Gesture Recognition in an IoT-Enabled System

Cheng-Ying Yang[a], Yi-Nan Lin[b], Sheng-Kuan Wang[b], Victor R.L. Shen[c,d], Yi-Chih Tung[b], Frank H.C. Shen[e], and Chun-Hsiang Huang[f]

[a]Department of Computer Science, University of Taipei 1, Taipei, Taiwan; [b]Department of Electronic Engineering, Ming Chi University of Technology 84, Taipei, Taiwan; [c]Department of Computer Science and Information Engineering, National Taipei University 151, New Taipei City TAIWAN; [d]Department of Information Management, Chaoyang University of Technology, Taichung, Taiwan; [e]Department of Electronic Engineering, Fu Jen Catholic University 510, Taipei, Taiwan; [f]Department of Electronic Engineering, Ming Chi University of Technology 84, New Taipei, Taiwan

ABSTRACT

Recently, with the vigorous development of the Internet of Things (IoT) technology, all kinds of intelligent home appliances in the market are constantly innovating. The public requirements for residential safety and convenience are also increasing. Meanwhile, with the improvement of indigenous medical technology and quality of life, people's average lifespan is gradually increasing. However, countries around the world are facing the problem of aging societies. Hand gesture recognition is gaining popularity in the fields of gesture control, robotics, or medical applications. Therefore, how to create a convenient and smart control system of home appliances for the elderly or the disabled has become the objective of this study. It aims to use Google MediaPipe to develop a hand tracking system, which detected 21 key points of a hand through the camera lens of a mobile device and used a vector formula to calculate the angle of the intersection of two lines based on four key points. After the angle of bending finger is obtained, users' hand gesture can be recognized. Our experiments have confirmed that the recognition precision and recall values of hand gesture for numbers 0–9 reached 98.80% and 97.67%, respectively; and the recognition results were used to control home appliances through the low-cost IoT-Enabled system.

ARTICLE HISTORY
Received 15 November 2022
Revised 28 January 2023
Accepted 31 January 2023

Introduction

Nowadays, with the rapid development of the Internet and technological products, people have ushered in an era when individuals are closely connected with each other. Moreover, many identification systems have been developed, such as sign language recognition, face recognition, and license plate recognition (Riedel, Brehm, and Pfeifroth 2021). However, there are still many flaws in hand gesture recognition. As their technology cannot recognize users' hand gestures quickly and accurately, it results in being unable to solve users' problems promptly

CONTACT Victor R.L. Shen rlshen@mail.ntpu.edu.tw Department of Computer Science and Information Engineering National Taipei University,New Taipei City 237, Taiwan

(Sharma et al. 2022). For this reason, it motivates us to find the resolution of all difficulties that seniors and people with disabilities may encounter at home, such as turning on and off the lights, locking the door, and making the phone calls.

This study is based on the concept of virtual touch system with assisted gestures by using deep learning and MediaPipe, which employed dynamic gestures to control the computer without using a mouse (Yu 2021). Moreover, the paper (Shih 2019) was also cited. In terms of complementary hardware and software requirements between these two works, the instant gesture tracking on wearable gloves was achieved by using the MediaPipe hand tracking module and calculating the angle of the intersection lines based on four key points. However, their method needs to further improve the accuracy and efficiency of recognition, which will be applied to more innovative cases.

How people can easily communicate with machines is now a new trend. Many researchers have tried to find reliable and humanized methods through the recognition of hand gestures, facial expressions, and body language, among which hand gesture is the most flexible and convenient one. Nevertheless, the hand tracking and recognition subjects are challenging due to the high flexibility of the hand (Riedel, Brehm, and Pfeifroth 2021).

This study aims to design a gesture recognition module incorporated into smart home systems so that both the elderly and the disabled people can control home appliances more comfortably and conveniently. Currently, everyone has a mobile device with camera lenses, which can identify the current gestures anytime and anywhere to control the home appliances (Gogineni, et al., 2020) when connected to Wi-Fi equipment. Therefore, it is such a novel device that all people can enjoy using (Alemuda 2017).

Problem Statement: In the existing methods, the integrity of an IoT-Enabled gesture recognition system has not yet been formally verified, and the soundness and feasibility of the model before system development cannot be ensured (Alemuda, et al., Alemuda 2017). Meanwhile, it is inconvenient and costly to wear a glove device (Shih, 2019). Furthermore, large volume and high-cost body sensing detection equipment cannot control home appliances, and the accuracy of American sign language (ASL) recognition is low (Lee 2019; Shen et al. 2022). Hence, this IoT-Enabled system is built to remedy the issues mentioned above.

Literature Review

This section presents an IoT-Enabled system for realizing smart control of home appliances by hand gesture recognition, which consists of MediaPipe and various development software tools, such as Thonny, Android Studio, and WoPeD.

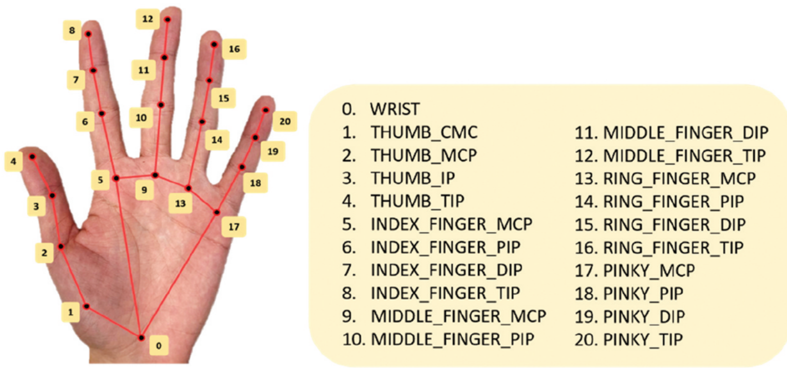

Figure 1. Positions of 21 hand key points.

MediaPipe Hand Tracking

MediaPipe is a development framework for processing machine learning (undefined). It can be applied to mobile devices, such as PC, Android, and iOS, by using the efficient management tools for CPU and GPU to achieve the goal of low latency. The perceptual pipeline can be constructed into modular graphics using MediaPipe, including model reasoning, media processing algorithms, and data conversion (Lugaresi et al. 2019).

Hand tracking is a key aspect of providing a natural way for people to communicate with computers. If the location of each key point on hand is found, the current gesture can be calculated by the angle of the finger, so that users can control the home appliances in the IoT-Enabled systems. MediaPipe hand has a function that tracks the hand and detects the hand markers such as palm and finger joints (Zhang et al. 2020). Through the machine learning algorithm, the hand position and markers can be found and inferred from a single frame.

After using MediaPipe hand tracking technology, 21 key points are located on the whole hand, as shown in Figure 1 (Ling et al. 2021). It is used to determine whether the hand is out of the recognition range or not. When the confidence level of recognition is below the default value, 0.5, the key point cannot be located, and the palm position requires to be tracked again.Figure 1 Positions of 21 hand key points.

The hand model is created with a dataset of 21 hand key points marked in a rectangular coordinate system. The model has three output values below.

(1) 21 key points of a hand, including the X-axis and Y-axis.
(2) Whether a hand is present in the image.
(3) Whether it is the left or right hand.

Control Board and Relay Module

The control board is the D1 mini ESP8266 microcontroller chip, which is a low-cost and low-power Wi-Fi microchip with full TCP/IP protocols. It has Wi-Fi connectivity, full hardware features, 32-bit microcontroller core, whose core frequency can be up to 160 MHz. Moreover, it can store data so that the old data can be read after a reboot. Meanwhile, it possesses 16 digital pins (GPIO) and one analog pin (ADC); and supports various protocols, such as UART, I2C, and SPI (Lin 2018). The ESP8266 microcontroller allows the control of external electronic components by using the input and output pins on both sides and to write Python programs using Thonny development environment.

Relay is an electronic control element whose internal circuit has two control systems, namely, the control circuit and the controlled circuit. Based on the reaction principle of small current controlling large one, it is often used in the automatic control system. Additionally, the relay resembles a switch with functions of automatic regulation, safety protection, and circuit conversion (undefined). Using this module with an ESP8266 microcontroller to connect home appliances, this study aims to control the home appliances through the hand tracking of MediaPipe.

Thonny Environment and Android Studio

As a Python development environment (GitHub Thonny 2022, 2022), Thonny is used to control the ESP8266 microcontroller. The top half is a part showing the code written, and the bottom half is the shell window which is used for discussion after program execution.

Android studio is the development environment for developing Android Apps (IDE) (GitHub Android, 2022; Get to, 2022). Each project in Android studio contains one or more modules with source code files and resource files. The module types include:

(1) Android application module,
(2) Program library module,
(3) Google App Engine module.

OkHttp is a third-party package for network connections, which is used to obtain network data. It has a more efficient connection with mechanisms such as unlinking and caching (OkHttp 2022; OkHttp Internet Connection 2022). To use OkHttp, one must additionally declare that the network is added and connected in the GRADLE (Module) level.

Petri Net

Petri net theory was developed by a German mathematician, Dr. Carl Adam Petri, which is basically a directed and mathematical graph of discrete parallel systems, suitable for modeling asynchronous and concurrent systems (Chen et al. 2021). It can be used to perform qualitative and quantitative analysis of a system, as well as to represent the systematic synchronization and mutual exclusion. Therefore, Petri net is widely used in different fields for system simulation, analysis, and modular construction (Hamroun et al. 2020; Kloetzer and Mahulea 2020; Zhu et al. 2019).

A basic PN model contains four elements, namely, *Place*, which is denoted as a circle; *Transition*, a long bar or square; *Arc*, a line with arrow; and *Token*, a solid dot, as listed in Table 1.

(1) *Place*: It represents the status of an object or resource in the system.
(2) *Transition*: It represents the change of objects or resources in the system. A transition may have multiple input and output places at the same time.
(3) *Arc*: It represents the transfer marker of objects in a system. The input place is connected to the output place through the transition, and the arrow represents the direction of the transfer.
(4) *Token*: A token represents a thing, information, condition, or object. When a transition represents an event, a place may or may not contain a token initially.

Petri nets are basically composed of three elements, $PN = (P, T, F)$, where

$P = \{p_1, p_2, \ldots, p_m\}$ denotes a finite set of places.

$T = \{t_1, \mathrm{t}_2, \ldots, t_m\}$ denotes a finite set of transitions.

$F = (P \times T) \cup (T \times P)$ denotes a set of lines with arrows (i.e. flow relation).

$M = \{m_0, m_1, m_2, \ldots\}$ denotes a set of markings. m_i denotes a vector in the set of M, representing the state of token distribution after the Petri net is triggered i times. Additionally, the value in a vector is an integer number indicating the number of tokens in the corresponding place.

Table 1. Four elements of Petri net.

Elements	Notations
Place	○
Transition	⟶ or ■
Arc	□
Token	●

WoPed Software

Workflow Petri Net Designer (WoPeD (Workflow Petri Net Designer), 2022; GitHub/WoPeD, 2022) is an open-source software tool developed by Cooperative State University Karlsruhe under the GNU Lesser General Public License (LGPL) that provides modeling, simulation, and analysis of processes described by workflow networks. WoPeD is currently maintained by Sourceforge (a web-based open-source development platform), and current development progress can be found on the home page of WoPeD project at Sourceforge. The verification of the system design process is carried out with this tool. The Petri net model is used to analyze the design process and to ensure the feasibility and soundness of a system.

Related Works

For some people with mobility problems who are unable to take care of themselves and need the help of others, or for some speakers who cannot use a mouse at a close distance, researchers C.R. Yu and F. Alemuda (Alemuda 2017; Yu 2021) proposed a method to use gesture recognition to control the actions of rolling up and down, the zooming in and out of the slides. However, the integrity and soundness of the systems have not yet been formally verified to ensure that the pre-development model of the system is feasible. The wearable glove in home appliances based on IoT technology was proposed by W.-H. Shih (Shih 2019), but it was found with inconvenience and high costs. Shih's experimental results indicate that his study employed deep images to recognize the user's gestures, which is like the method proposed by X. Shen (Shen et al. 2022). However, the experiments revealed that the body sensing devices were expensive, and their proposed methods were unable to control home appliances. Moreover, the precision of the American sign language (ASL) recognition was compared. The method of gesture recognition for letters A-Z proposed by S. Padhy (Padhy 2021) was tested, but the numbers were not yet recognized. Amazon has been a trendsetter through its Alexa-powered devices. Alexa is an intelligent personal assistant (IPA) that performs tasks, such as playing music, providing news and information, and controlling smart home appliances. A relationship between Alexa and consumers with special needs is established as it helps them regain their independence and freedom (Ramadan, Farah, and El Essrawi 2020). Recent improvements of the IoT technology are giving rise to the explosion of interconnected devices, empowering many smart applications. Promising future directions for deep learning (DL)-based IoT in smart city environments are proposed. The overall idea is to utilize the few available resources more smartly by incorporating DL-IoT (Rajyalakshmi, et al. Rajyalakshmi and Lakshmanna 2022).

Proposed Approach

In this section, hardware/software configurations, system structure, gesture recognition, and hand gesture definitions are presented.

Hardware and Software Configurations

ESP8266 microcontroller and relay module are selected. The hardware configuration can send high or low voltage signal to the D5 pin of the microcontroller through Wi-Fi, and then use the base voltage of a transistor to control the relay. The control mode leads the base voltage to send high voltage signal to energize the relay. In contrary, low voltage signal leads to relay disconnection so that the purpose of switching the home appliances off can be achieved. Hereby, the ESP8266 microcontroller is combined with RGB LED light bar. With the red line being 5 V, the brown line being GND, and the white line being D2 pin, the connection is thus completed.

System Structure

To confirm the system design flow by the combination of software with hardware, the execution sequence is converted into a flowchart, as shown in Figure 2 After opening the App, the detection model can locate the palm area in the image, and the hand area recognition model is able to mark the key points in the locked area. Finally, the gesture recognition system is used to mark the location points. Based on the angle of two joint lines, whether each finger is straight or bent can be judged, and the recognition results can be output. Given the corresponding command, users will thus control the home appliances through Wi-Fi equipment.

Gesture Recognition

When the camera is turned on, the user makes a gesture so that the system can capture 21 key points of a hand. For example, the gesture of number 1 uses four key points of 0, 6, 7, and 8 on the finger and palm. Two lines are formed as shown in Figure 3 The calculation of the angle of bending fingers is done by using the following formulas (Reference, 2022), where $Xa = x_2 - x_1$ and $Ya = y_2 - y_1$, $X_b = x_4 - x_3$, and $Y_b = y_4 - y_3$ transform into vectors $L1 = <X_a, Y_a>$ and $L2 = <X_b, Y_b>$ to find the inner product of two vectors $L1 \cdot L2 = X_a \times X_b + Y_a \times Y_b$, as listed in Table 2. Furthermore, based on Eq. (1) of inner product of two vectors:

$$\cos A = \frac{L1 \cdot L2}{|L1| \cdot |L2|} \tag{1}$$

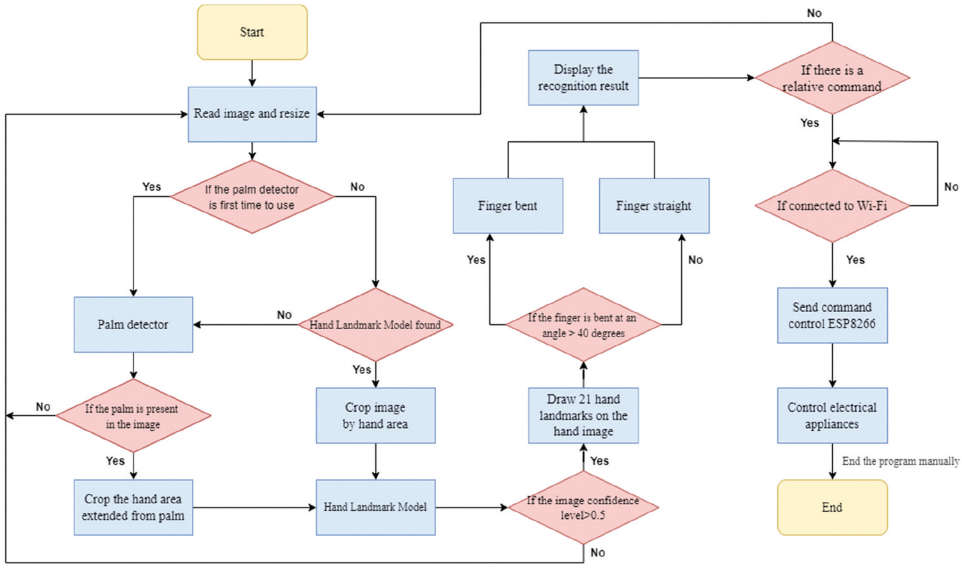

Figure 2. System operation flowchart.

where A denotes the angle between two vectors, and Eq. (2):

$$\cos A = \frac{(X_a \cdot X_b + Y_a \cdot Y_b)}{\sqrt{(X_a{}^2 + Y_a{}^2) \cdot (X_b{}^2 + Y_b{}^2)}} \tag{2}$$

the inverse trigonometric function is used to find the angle as Eq. (3):

$$A = \cos^{-1}\frac{(X_a \cdot X_b + Y_a \cdot Y_b)}{\sqrt{(X_a{}^2 + Y_a{}^2) \cdot (X_b{}^2 + Y_b{}^2)}} \tag{3}$$

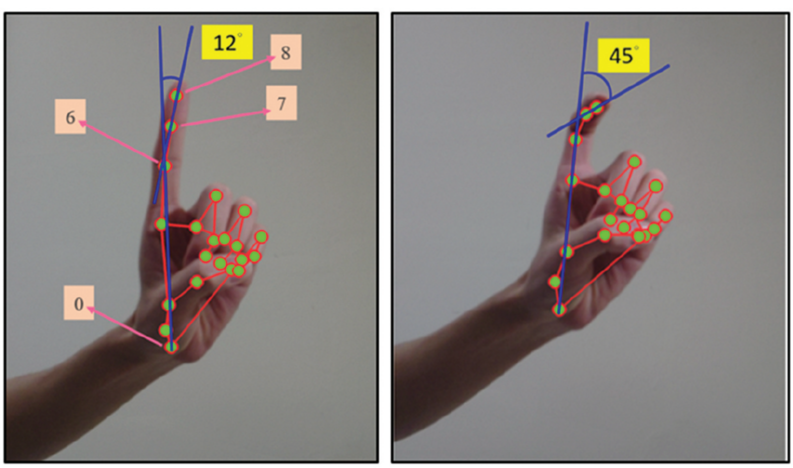

Figure 3. Different angles of gesture 1.

Table 2. Parameters of the fingers.

Parameters	Index finger	Middle finger	Ring finger	Little finger	Thumb
X_a	Key point 0	Key point 0	Key point 0	Key point 0	Key point 0
Y_a	Key point 6	Key point 10	Key point 14	Key point 18	Key point 2
X_b	Key point 7	Key point 11	Key point 15	Key point 19	Key point 3
Y_b	Key point 8	Key point 12	Key point 16	Key point 20	Key point 4
x_1	X coordinate of key point 0	X coordinate of key point 0	X coordinate of key point 0	X coordinate of key point 0	X coordinate of key point 0
x_2	X coordinate of key point 6	X coordinate of key point 10	X coordinate of key point 14	X coordinate of key point 18	X coordinate of key point 2
y_1	Y coordinate of key point 0	Y coordinate of key point 0	Y coordinate of key point 0	Y coordinate of key point 0	Y coordinate of key point 0
y_2	Y coordinate of key point 6	Y coordinate of key point 10	Y coordinate of key point 14	Y coordinate of key point 18	Y coordinate of key point 2
x_3	X coordinate of key point 7	X coordinate of key point 11	X coordinate of key point 15	X coordinate of key point 19	X coordinate of key point 3
x_4	X coordinate of key point 8	X coordinate of key point 12	X coordinate of key point 16	X coordinate of key point 20	X coordinate of key point 4
y_3	Y coordinate of key point 7	Y coordinate of key point 11	Y coordinate of key point 15	Y coordinate of key point 19	Y coordinate of key point 3
y_4	Y coordinate of key point 8	Y coordinate of key point 12	Y coordinate of key point 16	Y coordinate of key point 20	Y coordinate of key point 4
L_1	Vectors for key points 0 and 6	Vectors for key points 0 and 10	Vectors for key points 0 and 14	Vectors for key points 0 and 18	Vectors for key points 0 and 2
L_2	Vectors for key points 7 and 8	Vectors for key points11 and 12	Vectors for key points 15 and 16	Vectors for key points 19 and 20	Vectors for key points 3 and 4
A	The angle between L1 and L2 vectors	The angle between L1 and L2 vectors	The angle between L1 and L2 vectors	The angle between L1 and L2 vectors	The angle between L1 and L2 vectors

The complexity of the proposed model is denoted as $O(m+n)$, where m denotes the time to correctly recognize hand gestures and n denotes the time to control the home appliances. For scalability, the number of hand gestures can be easily scaled up by using the additional combination of key points of a hand.

When users make a gesture, the angle is obtained by the vector angle formula and the bending of a finger is determined. The lines connecting points 0 and 6 on the palm with points 7 and 8 on the index finger indicates gesture 1, as shown in Figure 3 The gesture has been tested on ten samples with various finger bending angles. Finally, 40-degree is determined as the best recognition standard. When the finger is straight, the angle obtained is 12 degrees; when the index finger is bent, the angle obtained is 45 degrees. Hence, 40-degree is set as the basis for determining whether a finger is bent or not. When the angle is more than 40 degrees, the finger is bent; and when it is less than 40 degrees, the finger is straight.Figure 3 Different angles of gesture 1.

Hand Gesture Definitions

Based on Eq. (3), the bending angle of each finger can be obtained so that number gestures 0–9 can be determined. If the angle of the intersection of two lines based on four key points is larger than 40 degrees, the output logic is 1; otherwise, the output is 0. When the output result is (0, 1, 0, 0, 0, 0), it indicates the state of each finger in order, as listed in Table 3 (thumb, index finger, middle finger, ring finger, and little finger), where the index finger is straight, and the other four fingers are bent. Therefore, it is determined that the user is making a gesture of number 1. In this study, ten gestures are thus defined.

System Verification and Experimental Results

This section presents the system verification and experimental results. Petri net of system design process and WoPeD software tool are utilized to model and analyze the simulation results. The experimental results of gestures under the conditions of different distances and unstable light sources, as well as the functional operation of numbers 1, 2, 4, 5, and 6, such as the control of light switch and the color of RGB light bar are all presented.

Petri Net Modeling

The Petri net modeling is an intuitive way to build a system framework (Gan et al. 2022; Lu et al. 2022) and to analyze and simulate it using WoPeD. The Petri net model is built for the analysis and simulation of the system design flowchart, as shown in Figure 4 The interpretation of places and transitions is listed in Tables 4 and 5, respectively. The design flow of the modeling is explained as follows: The start system of the PN model is represented by the initial marking of the place p_1 containing one token. This enables the firing of transition t_1 which moves the token from place p_1 to place p_2. In other words, p1 system preparation starts, where t1 (start) means the action of completing t1. Then, the token would be transmitted from p1 to p2 (prepare to capture the image and adjust image size) by t1 and start the action of t2 (cell phone lens captures the image and adjusts the image size). When the token reaches p3, it arrives at p4 through t3. If t3 (successfully use the palm detector for the first time) goes to t5 (palm detected) through p4 (execute palm detector), p7 (confirm palm capture) enters t9 (palm presents in the image), and then enters t10 (crop hand area extending from palm) through p8 (confirm palm's presence in the image). Instead, t8 (palm does not present in the image) returns to p2. If (unsuccessful for the first-time using palm detector) t4 enters the hand model through p5 (palm detector did not execute), it would be judged if the hand is found. If t6 (hand is not found) fires, it returns to t5 (palm detected) through p4 (palm position detected). On the contrary, if (hand is found), t7 fires to go through p6 (confirm hand model), enters t11 (crop the image according to the previous hand area), passes through p9 (confirm hand position), and goes to t12 (build hand model) through p10 (the value of hand model on confidence level). If t14 (hand in image confidence level<0.5) fires, then it returns to p2; otherwise, it goes to t13 (hand in image confidence level>0.5) through p11 (confirm image confidence level>0.5); and

Table 3. Ten gestures.

Numbers	Gestures	Left-hand finger states (thumb, index finger, middle finger, ring finger, little finger)
0		(0, 0, 0, 0, 0)
1		(0, 1, 0, 0, 0)
2		(0, 1, 1, 0, 0)
3		(0, 1, 1, 1, 0)
4		(0, 1, 1, 1, 1)

(*Continued*)

Table 3. (Continued).

Numbers	Gestures	Left-hand finger states (thumb, index finger, middle finger, ring finger, little finger)
5		(1, 1, 1, 1, 1)
6		(1, 0, 0, 0, 1)
7		(1, 1, 0, 0, 0)
8		(1, 1, 1, 0, 0)
9		(1, 1, 1, 1, 0)

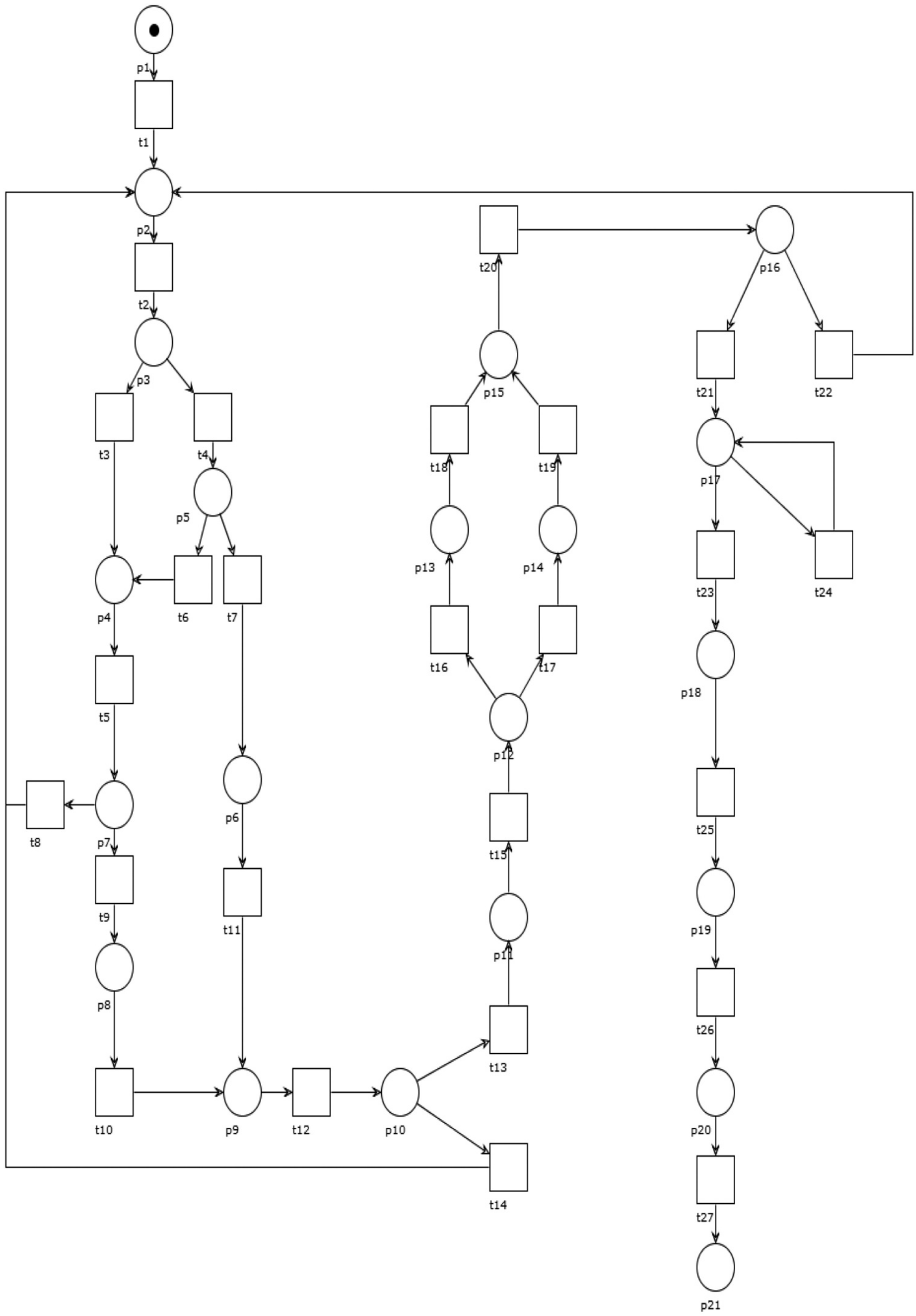

Figure 4. Petri net model of system design process.

Table 4. Interpretation of places.

Place	Interpretation	Place	Interpretation
p1	System preparation starts	p12	Take the intersection line of four key points to calculate the angle
p2	Prepare to capture the image and adjust image size	p13	Calculated finger angle > 40 degrees
P3	Execute palm detector	p14	Calculated finger angle < 40 degrees
p4	Confirm the detection of hand	p15	Confirm finger angle
p5	Failure in confirming palm's position in the first attempt	p16	Compare gesture definition sources
p6	Confirm hand model	p17	Matching with the corresponding command
p7	Confirm palm capture	p18	Matching with Wi-Fi successfully
p8	Confirm palm's presence in image	p19	Compare gesture to command data
p9	Confirm hand position	p20	Confirm the status of home appliances
p10	The value of hand model on a confidence level	p21	The end of system design process
p11	Confirm image confidence level >0.5		

Table 5. Interpretation of transitions.

Transition	Interpretation	Transition	Interpretation
t1	Start	t15	Output 21 hand key point images
t2	Smartphone lens captures image and adjusts image size	t16	Finger bent > 40 degrees (Yes)
t3	First time using palm detector (Yes)	t17	Finger bent > 40 degrees (No)
t4	First time using palm detector (No)	t18	Finger bent
t5	Palm detected	t19	Finger straightening
t6	Hand is found (No)	t20	Display gesture recognition result
t7	Hand is found (Yes)	t21	Finger has the corresponding finger (Yes)
t8	Palm presents in the image (No)	t22	Finger has the corresponding finger (No)
t9	Palm present in the image (Yes)	t23	Wi-Fi connection (Yes)
t10	Crop hand area extending from palm	t24	Wi-Fi connection (No)
t11	Crop the image according to the previous hand area	t25	Send the command to ESP8266
t12	Build hand model	t26	Control home appliances
t13	Hand in image confidence level > 0.5 (Yes)	t27	The end
t14	Hand in image confidence level > 0.5 (No)		

goes to t15 (output 21 hand key point images) through p12 (take the intersection line of four key points to calculate the angle). If (finger bend>40 degrees) t16, it goes through p13 (calculated finger angle>40 degrees) to t18 (finger bend). On the contrary, t17 (finger bend<40 degrees) fires to go through p14 (calculated finger angle<40 degrees) to t19 (finger straightening), passes through p15 (confirm finger angle) to fire t20 (display gesture recognition result) through p16 (compare gesture definition sources). If t22 (finger has no corresponding finger) fires, then it goes back to p2; otherwise, it goes to fire t21 (finger has the corresponding finger) and enters t24 (no Wi-Fi connection) through p17 (matching with the corresponding command). It then goes back to p17 (matching with the corresponding command). Otherwise, it goes to t23 (Wi-Fi connection) through p18 (matching with Wi-Fi successfully) and enters t25 (sending the command to ESP8266). It fires t26 (control electrical appliances) through p19 (compare gesture to command data). Through p20 (confirm the status of electrical appliances), it fires t27 (end) and finally goes through p21 (the end of system design process).

To verify the correctness of the system design process including hardware and software components, a workflow diagram is loaded into this program. Furthermore, this study has used 21 places as listed in Table 4 and 27 transitions as listed in Table 5.

System Verification

Figure 5 depicts the net statistics and the structural analysis of the PN model, which displays the total number of elements in the model and the soundness of the system design process. Consequently, there are no conflicts or deficiencies in the operational process, and the feasibility of the system is fully verified.

Experimental Results

Once the mobile device is connected to the ESP8266 microcontroller, it sends the command to the home appliances after completion of hand gesture recognition and the execution time back to the mobile device. After testing, it takes 0.62 seconds from making a correct gesture to turn on the LED light. The original LED light is turned on, but when the finger makes the gesture of number 1, the command makes the LED light turn off, as shown in Figure 6 Originally, the LED light turned off, when the finger makes the gesture of number 2, the command makes the LED light turn on, as shown in Figure 7 Therefore, this test shows that it is possible to use gesture recognition to control the home appliances.

As shown in Figure 8, there are 15 controllable light beads on the RGB light bar, and gestures can be used to make the light beads display according to the desirable brightness and color. When the finger makes the gesture of number 4, the command makes the RGB light bar turn red.

As shown in Figure 9, when the finger makes the gesture of number 5, it sends a command to make the RGB light bar turn blue.

As shown in Figures 10–12, when the finger makes the gesture of number 6, the command makes the red RGB light bar change the degree of brightness in three steps, namely, normal, slightly dim, and slightly bright.

In this study, ten samples were asked to make number gestures that could be judged by the naked eye in bright light with a simple background. They performed ten defined gestures at different angles, with the palms facing outward or inward, in a total of five movement patterns.

This might be due to the differences in gesture habits and the finger skeletal and muscular structures of each sample, resulting in differences of gesture movements. The precision value of each type of gesture in the MediaPipe model is listed in Table 6, and the recall value is listed in Table 7. The precision

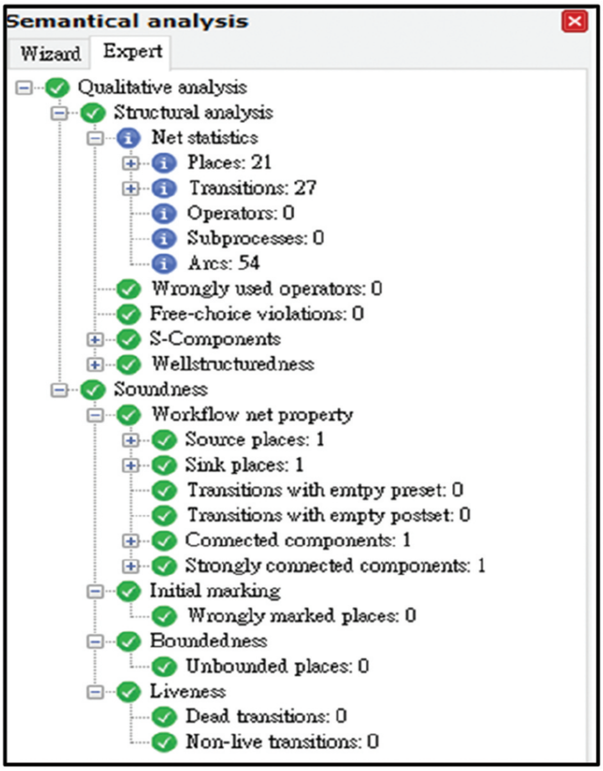

Figure 5. Semantical analysis.

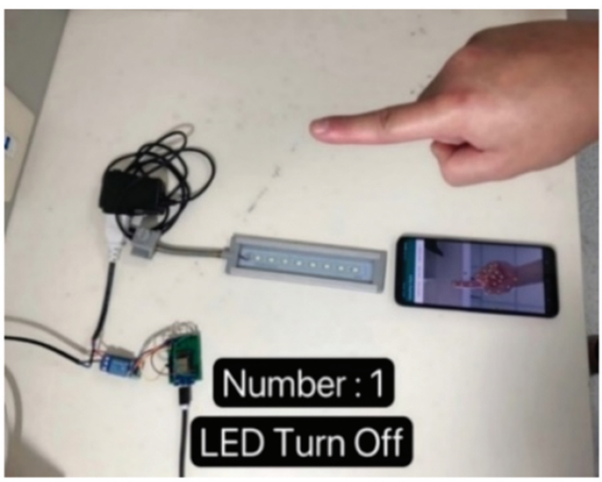

Figure 6. Gesture recognition - Light off.

calculation method Eq. (4) and the recall calculation method Eq. (5) are shown as follows:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{4}$$

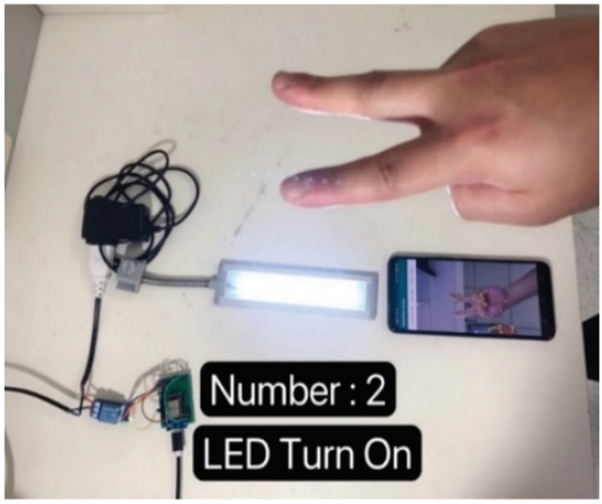

Figure 7. Gesture recognition - Light on.

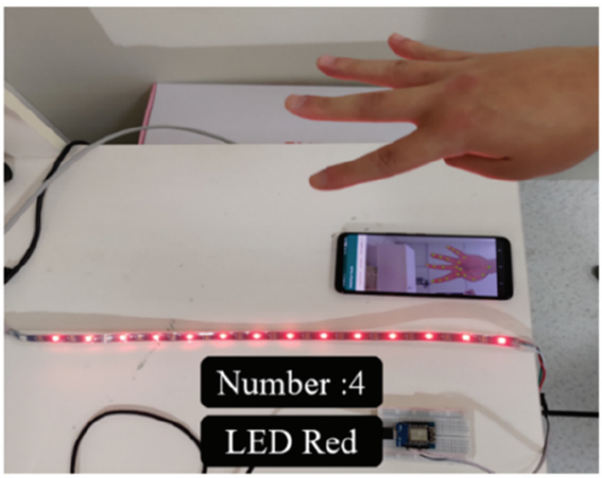

Figure 8. Gesture recognition –red RGB light bar.

$$\text{Recall} = \frac{TP}{TP+FN} \tag{5}$$

T (True) means the model recognition is correct.
F (False) means the model recognition is wrong.
P (Positives) means the model recognition is positive.
N (Negatives) means the model recognition is negative.
TP (True Positives) means the model recognition (positive) is the same as the actual result.
FP (False Positives) means the model recognition is different from the actual result.

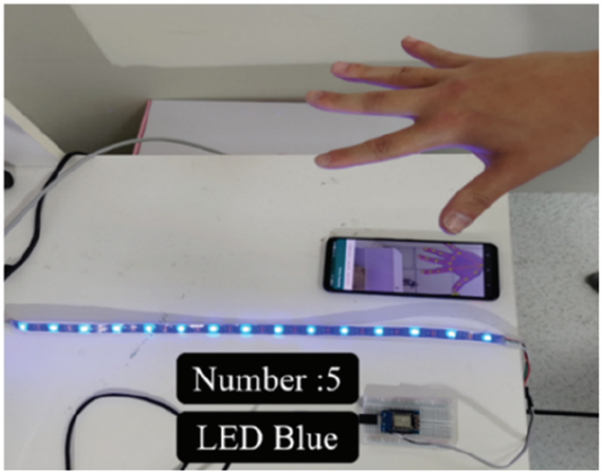

Figure 9. Gesture recognition –blue RGB light bar.

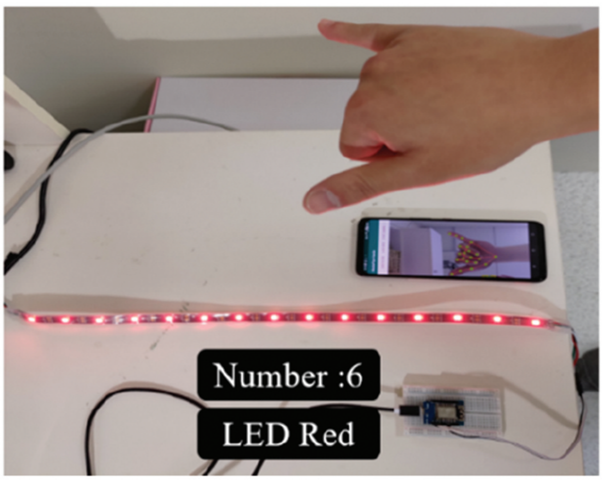

Figure 10. Red RGB light bar (normal).

FN (False Negatives) means the model recognition (negative) is different from the actual result.

The gesture recognition in the experiment is based on the American sign language (ASL) number gestures from 0 to 9. Ten samples were asked to make gestures with their fingers pointing upwards which could be detected by the naked eyes under bright light condition and with a simple background. One picture of each gesture is taken for identification, and the precision is listed in Table 8. This experiment refers to the multi-scenario gesture recognition using Kinect (Shen et al. 2022) and a deep image-based fuzzy hand gesture recognition method (Riedel, Brehm, and Pfeifroth 2021). The gestures of numbers 0, 1, 2, 4, 5, and 9 achieved 100% recognition success, with a total average

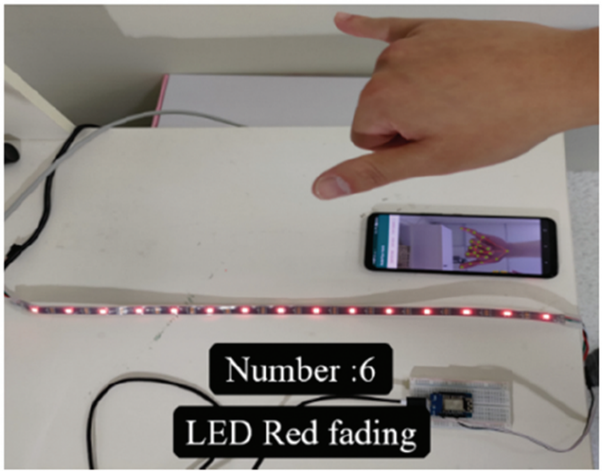

Figure 11. Red RGB light bar (slightly dim).

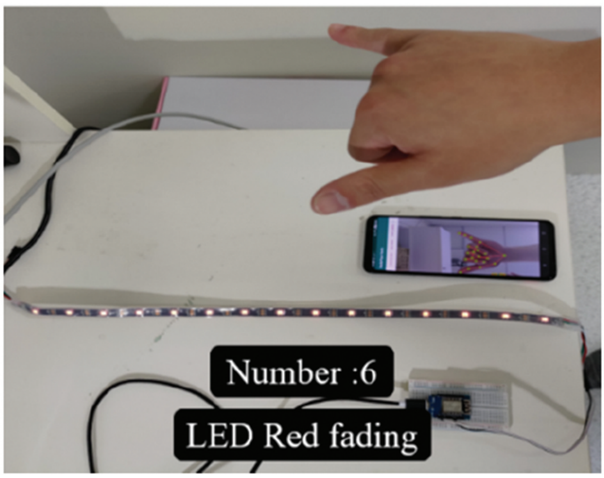

Figure 12. Red RGB light bar (slightly bright).

Table 6. Recognition results - average precision of 10 gestures.

Gesture	*TP*	*FP*	Precision	Average precision
0	50	0	1	0.988
1	50	0	1	
2	50	0	1	
3	50	0	1	
4	50	0	1	
5	50	0	1	
6	47	3	0.94	
7	50	0	1	
8	50	0	1	
9	47	3	0.94	

Table 7. Training results – average recall of 10 gestures.

Gesture	*TP*	*FN*	Recall	Average Recall
0	50	3	0.943	0.9767
1	50	0	1	
2	50	0	1	
3	50	0	1	
4	50	0	1	
5	50	0	1	
6	47	0	1	
7	50	0	1	
8	50	3	0.943	
9	47	0	1	

precision of 96%. However, the gestures of numbers 6, 7, and 8 are prone to recognition errors. The reason is that it may not be easy to judge the bending. The bending of the ring finger and middle finger affect other straight fingers. When the fingers are not straight, there are chances that they might be misjudged.

The best hand gesture recognition is performed when gestures are presented in a bright and open space. The distances of 1 m, 2 m, 2.5 m, and 3 m are selected as the test conditions. The test results show that gestures can be clearly recognized when the distances are 1 m, 2 m, and 2.5 m. However, when the test distance is 3 m, this system fails to recognize the gesture.

The distance test for gesture recognition was performed at a poor illumination of about 2.95 lux to test the recognition system. A total of 100 tests were conducted each at distances of 1 m, 2 m, and 3 m; and the successful recognition rate is only 32% because the recognition status is unstable at 1 m. During the tests at 2 m and 3 m distances, the light is not bright enough and the distance is too far away, which makes the recognition unsuccessful.

Taking gesture 2 as an example, the gesture is recognized normally at 0 degree in front view and 45 degrees on the side view. However, due to the angle difference, when the hand gesture is turned to 90 degrees, the finger is blocked, resulting in a wrong gesture recognition. The hand position and key points can still be captured, but the number recognition result is wrong.

Table 8. Comparison of the precision with different methods for gestures in ASL.

Methods Gestures	Proposed (%)	Shen, X. et al. (%)	Chen, Y.-C. (%)
0	100	95.27	100
1	100	91.75	98
2	100	89.25	95
3	90	90	98
4	100	100	95
5	100	100	100
6	90	77	91
7	90	84.25	91
8	90	76.25	83
9	100	74	96
Average	96	86.94	94.4

Table 9. Comparison of this study with other methods.

Methods Metrics	Proposed	Yu, C.-R.	Shih, W.-H.	Padhy, S.	Chen, X., et al.
Hardware	Smartphone, ESP8266 microcontroller chip	USB 3.0 bus; At least 4GB RAM webcam or smartphone	Wearable smart gloves, Arduino YUN	Surface electromyography (sEMG), Multilinear singular value decomposition (MLSVD)	Surface electromyography (sEMG)-based, CNN+LSTM (long short-term memory)
Software	MediaPipe Hand Android Studio Thonny 3.7.5	Windows 10 operating system, Python 3.6	Arduino operating system	Tensor-based approach, Dictionary Learning (DL)	Transfer learning (TL) strategy, CNN-based source network
Number of gestures	10	7	4	10	20
Control of the actions	1. All home appliances switch on/off 2. Control of light color	1. PPT files zoom in and zoom out 2. Slides playing, next page, and close	1. Power switch of light bulb, color and brightness setting 2. Power setting of air conditioner, temperature, and wind speed	(1) Upper limb motion classification (2) Biomedical engineeringsystems	(1) Myoelectric control systems (2) Biomedical engineeringsystems
Distance	2.5m	1.9m	2.0m	0.0m	0.0m
The success rate when the light illumination is 2.95 lux	32%	22%	25%	28%	29%
Response time(sec.)	0.62	0.87	0.91	0.90	0.89
Precision %	98.80	90.65	91.71	92.12	93.32
Recall %	97.67	89.11	90.16	91.23	92.77
System validation tool	WoPeD	N/A	N/A	N/A	N/A

N/A: Unavailable

Functional Comparison

A comprehensive comparison of this study with other methods is listed in Table 9. For hardware, the low-cost ESP8266 microcontroller was used; and for software, the MediaPipe hand tracking system developed by Google was used. In addition, Android Studio with gesture recognition was employed. The number of defined gestures is up to 10, the response time is about 0.62 seconds, the precision is 98.80%, and the recall is 97.67%. This system was also tested at a distance and at a poor illumination level of approximately 2.95 lux. Additionally, this system was modeled and analyzed using Petri net software tool, WoPeD, to ensure its integrity and soundness (Zeng et al. 2022). In summary, our system outperforms others in terms of different performance metrics.

Conclusion

A low-cost ESP8266 microcontroller chip is used to enable hand gesture recognition for smart control of home appliances. This study aims to use MediaPipe hand tracking to extend the fingers based on the palm and add the vector angle formulas to calculate the finger angle. Ten hand gestures were defined. To the end, the built system can control the home appliances via hand gestures with promising precision and recognition speed. This study has made the following contributions:

(1) The system design framework is modeled and analyzed using Petri net tool, WoPeD, to ensure its integrity and soundness. If the system has no errors, then it will accelerate the system production.
(2) It is easy and fast to operate, and the manufacturing cost is low, which only takes about 0.62 seconds to control home appliances.
(3) The vector formula was used to determine the bending angle of fingers to effectively improve the recognition of numbers 0-9, with the precision and recall values as high as 98.80% and 97.67%, respectively. The precision value in ASL reaches 96% when compared to other methods.

This system can help users operate the home appliances in a comfortable and convenient way. For example, there is no need for users to get up to switch on and off the home appliances. All they need to do is to connect mobile devices to Wi-Fi equipment to complete the actions of switching on and off the electric power.

In addition to controlling the home appliances, this study expects to be applied to medical or automotive related products. It is also anticipated that the results of this study will inspire more researchers to delve into the development of gesture recognition systems and create more innovative ideas. In this way, the public may enjoy the convenience brought by hand gesture recognition in the future.

Acknowledgements

The authors are grateful to the anonymous reviewers for their constructive comments which have improved the quality of this paper.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the Ministry of Science and Technology, Taiwan, under grants MOST 110-2637-E-131-005-, MOST 110-2221-E-845-002- and MOST 111-2221-E-845-003-.

References

Relay. [Online] Available: https://tutorials.webduino.io/zh-tw/docs/basic/component/relay.html. May 2022a.

Alemuda, F. 2017. Gesture-based control in a smart home environment, *Master Thesis*, International Graduate Program of Electrical Engineering and Computer Science,National Chiao Tung University.

Chen, X., Y. Li, R. Hu, X. Zhang, and X. Chen. 2021. Hand gesture recognition based on surface electromyography using convolutional neural network with transfer learning method. *IEEE Journal of Biomedical and Health Informatics* 25 (4):1292–304. doi:10.1109/JBHI.2020.3009383.

Gan, L., Y. Liu, Y. Li, R. Zhang, L. Huang, and C. Shi. 2022. Gesture recognition system using 24 GHz FMCW radar sensor realized on real-time edge computing platform. *IEEE Sensors Journal* 22 (9):8904–14. doi:10.1109/JSEN.2022.3163449.

Get to know android studio. [Online] Available: https://developer.android.com/studio/intro?hl=zh-tw. May 2022b.

GitHub android studio. [Online] Available: https://github.com/android. May 2022c.

GitHub Google/mediapipe. [Online] Available: https://github.com/google/mediapipe. May 2022d.

GitHub tfreytag/WoPeD. [Online] Available: https://github.com/tfreytag/WoPeD (Visited in 2022/05)

GitHub Thonny. [Online] Available: https://github.com/thonny/thonny. May 2022e.

Gogineni, K., A. Chitreddy, A. Vattikuti, and N. Palaniappan. 2020. Gesture and speech recognizing helper bot. *Applied Artificial Intelligence* 34 (7):585–95. doi:10.1080/08839514.2020.1740473.

Hamroun, A., K. Labadi, M. Lazri, S. B. Sanap, V. K. Bhojwani, and M. V K. 2020. Modelling and performance analysis of electric car-sharing systems using Petri nets. *E3S Web of Conferences* 170 (3001):1–6. doi:10.1051/e3sconf/202017003001.

Kloetzer, M., and C. Mahulea. 2020. Path planning for robotic teams based on LTL specifications and petri net models. *Discrete Event Dynamic Systems* 30 (1):55–79. doi:10.1007/s10626-019-00300-1.

Lee, F. N. 2019. A real-time gesture recognition system based on image processing, *Master Thesis*, Department of Communication Engineering, National Taipei University.

Lin, C. H. 2018. ESP8266-based IoTtalk device application: Implementation and performance evaluation, *Master Thesis*, Graduate Institute of Network Engineering, National Chiao Tung University.

Ling, Y., X. Chen, Y. Ruan, X. Zhang, and X. Chen. 2021. Comparative study of gesture recognition based on accelerometer and photoplethysmography sensor for gesture interactions in wearable devices. *IEEE Sensors Journal* 21 (15):17107–17. doi:10.1109/JSEN.2021.3081714.

Lugaresi, C., J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C. L. Chang, M. G. Yong, J. Lee et al, 2019. Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172. doi:10.48550/arXiv.1906.08172.

Lu, X., S. Sun, K. Liu, J. Sun, and L. Xu. 2022. Development of a wearable gesture recognition system based on two-terminal electrical impedance tomography. *IEEE Journal of Biomedical and Health Informatics* 26 (6):2515–23. doi:10.1109/JBHI.2021.3130374.

OkHttp Internet Connection. [Online] Available: https://ithelp.ithome.com.tw/articles/10188600. May 2022.

OkHttp : Use Report for Android Third-party. [Online] Available: https://bng86.gitbooks.io/android-third-party-/content/okhttp.html. May 2022.

Padhy, S. 2021. A tensor-based approach using multilinear SVD for hand gesture recognition from sEMG signals. *IEEE Sensors Journal* 21 (5):6634–42. doi:10.1109/JSEN.2020.3042540.

Rajyalakshmi, V., and K. Lakshmanna. 2022. A review on smart city - IoT and deep learning algorithms, challenges. *International Journal of Engineering Systems Modelling and Simulation* 13 (1):3–26. doi:10.1504/IJESMS.2022.122733.

Ramadan, Z., M. F. Farah, and L. El Essrawi. 2020. Amazon.Love: How Alexa is redefining companionship and interdependence for people with special needs. *Psychology & Marketing* 10 (1):1–12.

Reference program for static gesture-image 2D method GitCode. [Online] Available: https://gitcode.net/EricLee/handpose_x/-/issues/3?from_codechina=yes. May 2022f.

Riedel, A., N. Brehm, and T. Pfeifroth. 2021. Hand gesture recognition of methods-time measurement-1 motions in manual assembly tasks using graph convolutional networks. *Applied Artificial Intelligence* 36 (1):1–12. doi:10.1080/08839514.2021.2014191.

Sharma, V., M. Gupta, A. K. Pandey, D. Mishra, and A. Kumar. 2022. A review of deep learning-based human activity recognition on benchmark video datasets. *Applied Artificial Intelligence* 36 (1):1–11. doi:10.1080/08839514.2022.2093705.

Shen, X., H. Zheng, X. Feng, and J. Hu. 2022. ML-HGR-Net: A meta-learning network for fmcw radar based hand gesture recognition. *IEEE Sensors Journal* 22 (11):10808–17. doi:10.1109/JSEN.2022.3169231.

Shih, W. H. 2019. Applying IoT and gesture control technology to build a friendly smart home environment, *Master Thesis*, Department of Computer Science & Information Engineering, Chung Hua University.

Thonny, Python IDE for beginners. [Online] Available: https://thonny.org/. May 2022.

WoPeD (Workflow Petri Net Designer). [Online] Available: https://woped.dhbw-karlsruhe.de/ (Visited in 2022/05)

Yu, C. R. 2021. Virtual touch system with assisted gesture based on deep learning and MediaPipe, *Master Thesis*, Department of Computer Science and Information Engineering, National Chung Cheng University.

Zeng, J., Y. Zhou, Y. Yang, J. Yan, and H. Liu. 2022. Fatigue-sensitivity comparison of sEMG and a-mode ultrasound based hand gesture recognition. *IEEE Journal of Biomedical and Health Informatics* 26 (4):1718–25. doi:10.1109/JBHI.2021.3122277.

Zhang, F., V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C. L. Chang, and M. Grundmann. 2020. MediaPipe hands: On-device real-time hand tracking. *arXiv preprintarXiv: 200610214*

Zhu, H., J. Chen, X. Cai, Z. Ma, R. Jin, and L. Yang. 2019. A security control model based on Petri net for industrial IoT, *Procs. of IEEE International Conference on Industrial Internet (ICII)*, Orlando, FL, USA, 156–59.