International Journal of Applied Science and Engineering

Special issue: The 10th International Conference on Awareness Science and Technology (iCAST 2019)

Integrating gesture control board and image recognition for gesture recognition based on deep learning

Rung-Ching Chen[1], Christine Dewi[1,3], Wei-Wei Zhang[1,2], Jia-Ming Liu[1], Su-Wen Huang[1,4*]

[1] Department of Information Management, Chaoyang University of Technology, Taiwan, R.O.C.
[2] Ningxia University, Xixia District, Yinchuan City, Ningxia
[3] Information Technology, Satya Wacana Christian University, Central Java, Indonesia
[4] Taichung Veterans General Hospital, Taiwan, R.O.C.

ABSTRACT

Due to the rise of the Internet of Things, more devices can connect with the Internet. A large amount of data is collected from devices that could be used for different applications. The development of hardware equipment for the Internet of Things not only use for industrial but also for smart homes. The smart home covers broad topics, including remote control of home applications, sensing of humans, temperature-controlling air conditions, and security monitors. When we carry out these topics, the human-machine interface is essential for system applications. A gesture recognition system is applied to many real applications. The reason is that the accuracy rate and real factors are complicated. They commonly use of gesture control service in the market is the sensor board of gesture control. The principle is using the electric field to change and determine the gestures. The limitation requires the close operation, and there is a problem of critical point sensitivity. In this paper, we use the gesture control board to combine with gesture image recognition methods to perform the double authentication gesture recognition. Raspberry Pi is the control center to integrate the intelligent light bulb. HUE makes a gesture recognition system. The results explain that the accuracy rate of the gesture recognition proposed is 90%. Meanwhile, it is higher than the SVM method.

Keywords: Raspberry Pi, Gesture recognition, Deep learning.

OPEN ACCESS

Received: March 29, 2020
Revised: June 6, 2020
Accepted: July 28, 2020

Corresponding Author:
Su-Wen Huang
dale33663366@gmail.com

Copyright: The Author(s). This is an open access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0), which permits unrestricted distribution provided the original author and source are cited.

Publisher:
Chaoyang University of Technology
ISSN: 1727-2394 (Print)
ISSN: 1727-7841 (Online)

1. INTRODUCTION

With the rapid development of the Internet and hardware, people have more interest in the research application of the Internet of Things (IoT) (Doan et al., 2016; Christidis and Devetsikiotis, 2016). IoT is connected to the items that would not otherwise be contacted through wired networks and wireless networks or Bluetooth, which could link devices into one or more networks so that hardware devices can send data to each other and execute instructions. Internet of Things has a comprehensive range of applications, for example, factories, environmental detection, and smart homes. The factory can install sensors on the devices, collect and statistics the data.

Moreover, when there are abnormal signs in the data, analysts will respond on time. Data detection can improve the process and increase production. Environmental detection is one of the research projects of the Internet of Things. It can save energy using Raspberry Pi and various sensors to collect environmental information from research labs and then using fuzzy logic to determine whether the status of the research room needs to switch lights and fans.

A smart home (Risteska and Trivodaliev, 2017; Sakshi and Radha, 2019) is also one of the categories of the Internet of Things. Many sensors are installed in the house, such as temperature, humidity sensors, CO_2 concentration sensors, and other input devices such as cameras. Furthermore, many sensors can collect information that has become the era of big data. Moreover, with many external tools to collect such a huge amount of data, we need a small processor. First, data collection, integration, and avoid clutter. The Raspberry Pi (Newmarch, 2017) is a mini-Linux (Linux kernel, 2020) computer developed by the British Raspberry Pi Foundation. There are many IO interfaces and a highly scalable modular design. It is quite suitable for marginal computing functions. After preprocessing the data, it can be uploaded to the server for calculating and returning the results displayed by the Raspberry Pi, and send instructions to other devices for the next action.

Nowadays, the most direct application of smart homes is the change in home situations and the users' experiences, such as lighting and playing music. All of those are focused on improving the quality of life through smart products. The man-machine interface has become very important. Before, the computer's input interface was a keyboard and a mouse. Now, there are many new methods, such as voice, gesture recognition (Kotsidou, 2018), and face recognition (Chen et al., 2019). They are more direct and user-friendly operation. Improving the user's operating mode and supporting a better experience in speech recognition or machine vision to collect features and then calculate the results. For example, speech recognition is often referred to as Apple's voice assistant (Siri) (Aron, 2011). It can control home appliances by dialogue with the Siri phone. Besides, committed to the development of artificial intelligence, Google (Dai, 2007) introduced OK Google (Kelly, 2012) also has a similar function.

Furthermore, the difficulty lies in need for speech analysis, semantic analysis, and the machine vision approach, generally by photography obtaining information, through feature extraction analysis. This type of analysis can be achieved in the past by a neural network and has good experimental results. Moreover, this paper uses deep learning in neural networks to conduct the experiment.

Regarding gesture recognition (Yu and Yuan, 2019; Li et al., 2020), there are several methods. First, extract information from a camera, such as RGB, grayscale, or a binarization matrix. However, because of changes in the environment causes misidentifications for recognition, such as brightness, shadows, especially when using a specific device, such as: wearing a sensing glove, reading the value on the sensor, and giving meaning. The equipment is expensive and inconvenient for the smart home environment of this article. There is no way to provide the user with a good experience. Another way is to use the gesture sensing board to change the electric field and get the information of the hand displacement. The function of gesture recognition is achieved, but this result has a limitation. The user must perform waving only at the designated position.

In summary, the objective of this paper hopes to improve the application of gesture recognition by using image and sensor board. Also, this paper will use the gesture control board to combine with gesture image recognition methods to perform the double authentication gesture recognition.

2. MATERIALS AND METHOD

2.1 Related works

The partially discussed in the literature will sort out relevant articles on smart houses and hand gesture recognition.

2.1.1 Smart home

The rise of the Internet of Things (IoT) (Gregorio et al., 2020) affected people's lifestyles. Many new mods are designed to make life more convenient. Smart homes give people a more comfortable living environment. For instance, if you work outside for a whole day, you want to relax and go home. You can automatically put in restful services such as relaxing music when entering the room, and smart homes often combine with some human behavior patterns to improve the application. In addition, the combination of face recognition can identify people's emotions and match the behavior patterns of some people before letting the machine make some decisions. The more human factors, the smart home can replace the advantages of traditional automated engineering. All IoT devices are connected using a control center. However, the selling price is often daunting, but Raspberry Pi can replace the control center. It has high compatibility, expandability, and the price is relatively low. The number of smart devices has rapidly expanded, and these devices have been integrated into our lives (Alanwar et al., 2017). The research direction of intelligent devices is mostly to improve the reliability, security of communications for machine learning and data analysis on the collected big data. People hope that Human-Computer Interaction (HCI) can be more direct and more straightforward. There are currently smart home service providers, such as FIBARO (2020). The system offers a variety of services such as remote control of home devices, active maintenance of a healthy environment, automatic adjustment of temperature and humidity, monitoring of air quality, home care, and home entertainment functions. Nowadays, the construction of smart homes is based on FIBARO smart home manufacturers (Buzzi et al., 2018). Users need a hosting control to take charge of the linkage of all devices. Linked devices are divided into four types: sensing devices such as water drop sensors, door and window sensors. Actuators such as loop controllers, lighting controllers. Operator interfaces such as remote controllers for scenes, gesture controllers, and smart. Integrated products such as APPLE TV, HUE LEDs, and camera.

2.1.2 Gesture recognition

Chung and Chiu (2014) mentioned that the hand is the most flexible part of the human body and will vary according to each person's habits. For instance, the time of hand gestures and spatial displacement. These individual characteristics can make a difference. The author applies his ideas to the access control system of smart homes. In this research, use hand displacement coordinates and depth image data of the hand to obtain hand trajectory and hand type data, using the algorithm of Histogram of Oriented Gradient (HOG) (Adetiba and Olugbara, 2015). After that, remove the hand features and use Support vector machines (SVM) (Chen et al., 2005; Dewi and Chen, 2019) to classify the results. However, there are many types of input images: dynamic gestures, continuous data, static gestures, general image characteristics, gesture tracking, past motion detection, and skin color detection (Zhu et al., 2002). The moving object detection method proposed by other scholars to detect gestures. Through the background subtraction method, the pixels of each frame of the continuous image can be subtracted to know the position of the moving object (Shi et al., 2010), and other scholars proposed to trace the face of a complex background through skin color. The method uses the RGB values to define the threshold value, to distinguish the skin color range and obtain the human face range and apply it to the gesture. It is the same principle. There is one important feature in gesture tracking that needs to be captured. In static gestures, there is no feature of moving trajectories. Displacements in dynamic gestures are often treated as an important feature (Wenjun et al., 2010). Scholars have mentioned that it is possible to use the shortest distance between the two frames of the hand, take two points out of the coordinates, and calculate the angle to get the eight-point map.

The above methods are based on features extracted from 2D cameras. Such practices often result in misjudgments due to changes in the environment, such as changes in light of the degree of background clutter, as suggested by Chong Wang (Wang et al., 2015). This research improves the use of the Kinect (Smisek et al., 2011) depth camera to obtain depth image information and expand the original 2D image data into 3D image data. The depth and the distance will not be affected by light changes and background clutter so that it can have higher adaptability.

Gesture recognition (Mitra and Acharya, 2007) methods need to solve all problems but how to define the start and end of gestures. In previous research, the use of background subtraction in Doan et al. (2016) and other researchers to get the beginning and end of the gesture reference point, but it can cause misjudgment in some situations. For example, in the chat, gestures are often used to assist attitudes or emotions when speaking, but obviously, there is no need to judge, but they are inadvertently executed.

2.1.3 System architecture

In real-life, many designs are double-checked for safety. For instance, the cutting machine needs a two-hand operation, and the dark lock of the car anti-theft requires one more action to start. The meaning of double confirmation is to make the user surer about the product quality. Determining the need to perform actions and reduce the chance of false touches. Applying this idea to gesture recognition is the main idea of this paper. The so-called double confirmation means that two different confirmation mechanisms are needed as a reference. The gesture recognition can be realized by means of wearable devices, gesture sensing boards, and images. Wearable devices are not practical for smart families and related gesture control (Haghi et al., 2017). As shown in Table 1, this paper proposes a gesture recognition system based on image combined with gesture sensor board. As shown in Fig. 1, using the electric field sensor board to detect whether there is a gesture that needs to be identified. The probability of reducing false positives is also achieved. Intuitive manipulation is required with any additional equipment.

This requires highly complex operations, such as servers, and sensors that receive data only need to be implemented. Simple actions such as obtaining data, when there are numerous sensors, a control center is needed to carry out equipment control and data collection to achieve the purpose of the division of labor.

2.1.4 Deep learning

This paper proposes an architecture that uses the Raspberry Pi as a control center to collect gesture features and process data to the server for deep learning prediction, and then return the prediction results, where the network camera and gesture sensor are input. Equipment; smart home objects such as smart bulbs are output devices, and

Table 1. Gesture recognition methods

Condition	Image recognition	Wearable device	Gesture control board
Whether the user needs to wear the device	N	Y	N
Accuracy	M	H	H
Convenience	H	L	M
Cost	L	H	L
Use filed restrictions	M	L	H

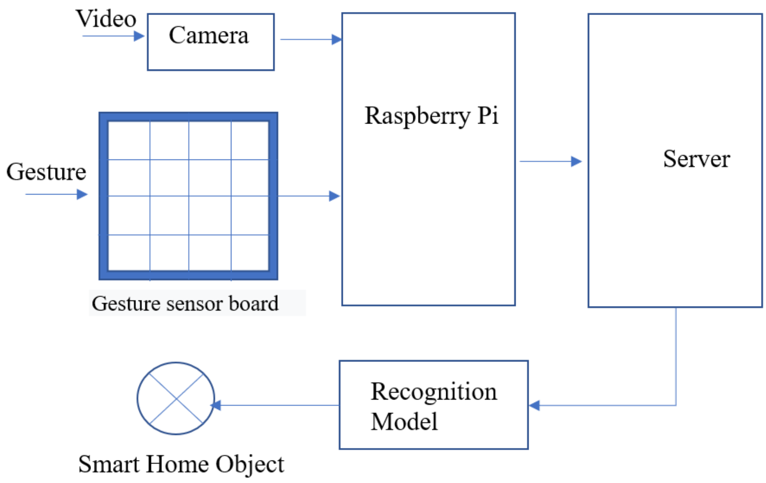

Fig. 1. The system structure.

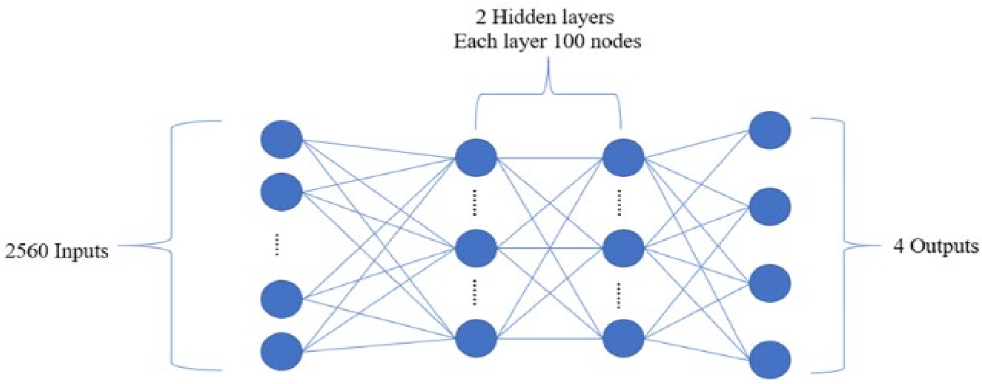

Fig. 2. The deep learning network architecture.

the server uses the H2O suite package in the R language for training and prediction. The neural network parameters of deep learning are shown in Table 2, and the network architecture is shown in Fig. 2. Fig. 2 shows the neural network architecture of the deep learning dimension supervised learning, in which the data processed by the Raspberry Pi image is used as an input, which is 2560 input point.

Table 2. Neural network parameters.

Hidden Layers	100,100
Epochs	100
Input nodes	2560

In deep learning, the excitation function is often used to converge the results. Common functions are Rectifier (1) and Tanh (2) (Kalman and Kwasny, 2003). This paper uses Rectifier and Tanh functions to perform experiments. The x represents the output of the deep learning network.

$$f(x) = \begin{cases} 0 & for\ x < 0 \\ x & for\ x \geq 0 \end{cases} \tag{1}$$

$$\tanh(x) = \frac{2}{1+e^{-2x}} - 1 \tag{2}$$

The capture of gesture features is the key to the gesture recognition process. In the past, most of the gesture recognition methods judged when a gesture starts or ends with a rare condition, which often causes misjudgment. For example, the camera usually can-do wrong judgment of gesture recognition. When we talk to someone, we often have gestures to enhance the emotions. This situation does not have gesture recognition that will be a misjudgment. Therefore, this paper proposes a gesture feature interception process, as shown in Fig. 3. Two different types of confirmation mechanisms are used to avoid system misjudgment. In this study, the method of judging the start of the gesture is to use the gesture sensor board to receive the signal and then determine whether the gesture starts. When the confirmation gesture begins, the Raspberry Pi launches the camera to capture the gesture image. In the present study, the method of determining the end of the gesture is the threshold value detection method. When the

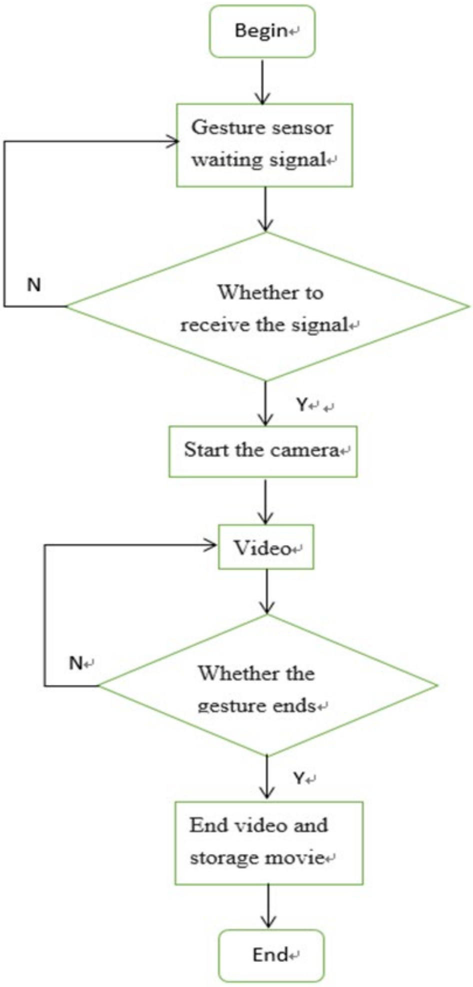

Fig. 3. Gesture feature capture flow chart.

threshold value is obtained, the first time the image is captured, the image is captured ten times, and the ten images are grayed out and then added to the matrix to obtain an average value of the non-gesture threshold. Next, the camera calculates the threshold value while recording; if the threshold of the current image is equal to the threshold value of no gesture, it is determined that the gesture ends and the recording ends.

Feature extraction in the gesture recognition process is a link-worthy of attention for gesture recognition (Verma and Salour, 2020). The quality of feature extraction will affect the accuracy of gesture recognition. The practice of this paper is to cut the recorded gesture film for each frame, and then use the method of extracting keyframes to obtain ten keyframes. Because the movie is composed of manystructures of continuous images, taking out keyframes can not only reduce the data dimension but also avoid over-learning. We will extract each frame of the gesture film obtained, and calculate the total number of frames, and then divide the total number of frames by ten to get the keyframe base. We can find the ten frames that are evenly distributed in the total number of frames. The frame is shown in Fig. 4. Also, because the size of the movie recorded by the camera is 640*480, if the pixel size is used as the feature input, the dimension will be too large and time-consuming.

To solve this problem, we will take out the ten keyframes and use the Raspberry Pi to perform image processing to reduce the dimension. The method adjusts the keyframe size to a size of 16*16 pixels as shown in Fig. 5, and converts it into a grayscale image, and then adjusts the two-dimensional matrix of the 16*16 pixels size of ten frames into a one-dimensional array. Fig. 6 describes a total of 2560 pixels fields. Using the above method, the gesture features can be extracted, as shown in Fig. 6.

3. RESULTS AND DISCUSSION

3.1 The experiments

This experiment will train a deep learning neural network and compare the experimental results for different activation functions, and in fact, make a complete system. The detailed contents will be divided into the following subsections to discuss: server setup, Raspberry Pi, and servo the devices exchange information and present results.

3.2 Environmental specifications

The server specification for this experiment is Intel(R) Core (TM) i7-4770 CPU @ 3.40 GHz, memory is 8.00 GB, the operating system is windows10, and the software for setting up the server is the Apache HTTP server. The version of the server software commonly used by the public is 2.4.17. In this system, there are functions to the PHP syntax for R execution and return data. This experiment used version 5.6.16. The software for deep learning in this experiment is the R language (R., 2011). The R version 3.4.1 released on 2017-06-30 and the package H20 is used for deep learning.

3.3 Data set

The training set collects up, down, left, right, turn-left, and turn-right gestures by the webcam for 100 strokes. The total of 600 data, each of which is 2560 fields. The test set data are collected by the webcam, up, down, left, right, turn-left, turn-right, and each as 20 test data has 2560 fields.

3.4 Evaluation

This paper uses Mean Square Error (MSE) in formula (3) (Chai and Draxler, 2014) and Root Mean Square Error (RMSE) in formula (4) to evaluate the model as good or bad. In formula (5), Ti is the predicate distance of each gesture and n is the number of gestures and the accuracy, in formula (6), is the summary of correct prediction of all gestures.

The results obtained from Table 3 show that the Tanh function performs better than the Rectifier function, so the experiment is mainly based on the Tanh function.

Fig. 4. Keyframe extraction.

pix1	pix2	…	pix15	pix16
pix17	pix18	…	pix31	pix32
…	…	…	…	…
pix225	pix226	…	pix239	pix240
pix241	pix242	…	pix255	pix256

Fig. 5. (16*16) pixel size.

pix1	pix2	pix3	…	pix254	pix255	pix256

Fig. 6. One-dimensional matrix.

$$MSE = \frac{1}{n} \sum_{t=1}^{n} (Actual\_value_t - prediction_t)^2 \tag{3}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (Actual\_value_t - prediction_t)^2} \tag{4}$$

$$Ti(aim, predict) = \begin{cases} 0 & if \ \ aim \neq predict \\ 1 & if \ \ aim = predict \end{cases} \tag{5}$$

$$\text{Accuracy} = \sum_{i=1}^{n} \frac{Ti}{n} \times 100\% \tag{6}$$

Table 3. Model evaluation

Evaluation	Rectifier	Tanh
MSE	1.969385e-08	**1.082136e-08**
RMSE	0.0001403348	**0.0001040258**

Chen et al., International Journal of Applied Science and Engineering, 17(3), 237–248

The formula for calculating the accuracy of this paper is as follows. When the predicted value is the same as the target value (aim), one is marked, and 0 is the opposite.

Table 5 is the prediction results on gestures, where the "Aim" is the target value and "Predict" is the predicted value. We can see seven errors in the twenty predictions. In Table 5, the values stand for the output of the probability of the detecting gesture belonging to the types of gestures. From Table 5, we can find the up gesture obtains the highest error rate among the six types of gestures.

Table 4. Confusion matrix: Row labels: Actual class; Column labels: Predicted class.

	Tanh-train						
	Down	Left	Right	Up	Turn left	Turn right	Error rate
Down	100	0	0	0	0	0	0.000 = 0/100
Left	0	100	0	0	0	0	0.000 = 0/100
Right	0	0	100	0	0	0	0.000 = 0/100
Up	0	0	0	100	0	0	0.000 = 0/100
Turn left	0	0	0	0	100	0	0.000 = 0/100
Turn right	0	0	0	0	0	100	0.000 = 0/100
Totals	100	100	100	100	100	100	0.000 = 0/600

Table 5. The Tanh function's prediction results in the up gesture.

No	Aim	Predict	Down	Left	Right	Turn left	Turn right	Up
1	Up	Up	0.00	0.00	0.00	0.00	0.00	1.00
2	Up	Up	0.00	0.00	0.00	0.00	0.00	1.00
3	Up	Up	0.00	0.00	0.00	0.00	0.00	1.00
4	Up	Up	0.00	0.00	0.00	0.00	0.00	1.00
5	Up	Up	0.00	0.00	0.00	0.00	0.00	1.00
6	Up	Up	0.00	0.00	0.00	0.00	0.00	1.00
7	Up	Up	0.00	0.00	0.00	0.00	0.00	1.00
8	Up	Up	0.00	0.00	0.00	0.00	0.00	1.00
9	Up	Up	0.00	0.00	0.00	0.00	0.00	1.00
10	Up	Up	0.00	0.00	0.02	0.00	0.00	0.98
11	Up	Up	0.00	0.00	0.34	0.04	0.00	0.61
12	**Up**	**Right**	**0.01**	**0.00**	**0.93**	**0.04**	**0.00**	**0.02**
13	**Up**	**Right**	**0.00**	**0.30**	**0.54**	**0.04**	**0.00**	**0.12**
14	**Up**	**Down**	**0.42**	**0.38**	**0.00**	**0.18**	**0.00**	**0.01**
15	Up	Up	0.00	0.15	0.14	0.07	0.00	0.64
16	**Up**	**left**	**0.01**	**0.36**	**0.27**	**0.18**	**0.00**	**0.19**
17	**Up**	**Turn Left**	**0.18**	**0.17**	**0.02**	**0.47**	**0.00**	**0.18**
18	**Up**	**Down**	**0.75**	**0.12**	**0.01**	**0.00**	**0.00**	**0.12**
19	**Up**	**Down**	**0.78**	**0.03**	**0.04**	**0.09**	**0.00**	**0.06**
20	Up	Up	0.16	0.07	0.09	0.04	0.00	0.63

Table 6 is the prediction result about the under gesture, in which the sixth judgment error and the gesture accuracy rate is 95%.

Table 7 shows the prediction result of the left gesture, whose accuracy rate is 100%. This is the only correct category in this experiment.

Table 6. The Tanh function's prediction results in the down gesture.

No	Aim	Predict	Down	Left	Right	Turn left	Turn right	Up
1	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
2	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
3	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
4	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
5	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
6	**Down**	**Left**	**0.13**	**0.86**	**0.00**	**0.00**	**0.00**	**0.01**
7	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
8	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
9	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
10	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
11	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
12	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
13	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
14	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
15	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
16	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
17	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
18	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
19	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00
20	Down	Down	1.00	0.00	0.00	0.00	0.00	0.00

Table 7. The Tanh function's prediction results in the left gesture.

No	Aim	Predict	Down	Left	Right	Turn left	Turn right	Up
1	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
2	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
3	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
4	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
5	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
6	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
7	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
8	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
9	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
10	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
11	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
12	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
13	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
14	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
15	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
16	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
17	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
18	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
19	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00
20	Left	Left	0.00	1.00	0.00	0.00	0.00	0.00

Table 8 shows the prediction result of the right gesture, in which the 10th data is wrong, and the accuracy rate is 95%.

Table 9 is the prediction result of the turn-left gesture, where the fifth predict is an error, and the accuracy rate is 95%.

Table 8. The Tanh function's prediction results in the right gesture.

No	Aim	Predict	Down	Left	Right	Turn left	Turn right	Up
1	Right	Right	0.00	0.00	0.98	0.00	0.02	0.00
2	Right	Right	0.00	0.00	1.00	0.00	0.00	0.00
3	Right	Right	0.00	0.00	1.00	0.00	0.00	0.00
4	Right	Right	0.00	0.00	1.00	0.00	0.00	0.00
5	Right	Right	0.00	0.00	1.00	0.00	0.00	0.00
6	Right	Right	0.00	0.00	1.00	0.00	0.00	0.00
7	Right	Right	0.00	0.00	1.00	0.00	0.00	0.00
8	Right	Right	0.00	0.00	1.00	0.00	0.00	0.00
9	Right	Right	0.00	0.00	1.00	0.00	0.00	0.00
10	**Right**	**Turn Right**	**0.00**	**0.00**	**0.18**	**0.00**	**0.82**	**0.00**
11	Right	Right	0.00	0.00	1.00	0.00	0.00	0.00
12	Right	Right	0.00	0.00	1.00	0.00	0.00	0.00
13	Right	Right	0.00	0.00	1.00	0.00	0.00	0.00
14	Right	Right	0.00	0.00	0.97	0.00	0.03	0.00
15	Right	Right	0.00	0.00	1.00	0.00	0.00	0.00
16	Right	Right	0.00	0.00	1.00	0.00	0.00	0.00
17	Right	Right	0.00	0.00	1.00	0.00	0.00	0.00
18	Right	Right	0.00	0.00	0.96	0.00	0.04	0.00
19	Right	Right	0.00	0.01	0.82	0.00	0.17	0.00
20	Right	Right	0.00	0.00	1.00	0.00	0.00	0.00

Table 9. The Tanh function's prediction results in the turn-left gesture.

No	Aim	Predict	Down	Left	Right	Turn left	Turn right	Up
1	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
2	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
3	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
4	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
5	**Turn Left**	**Left**	**0.00**	**0.98**	**0.00**	**0.02**	**0.00**	**0.00**
6	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
7	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
8	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
9	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
10	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
11	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
12	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
13	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
14	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
15	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
16	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
17	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
18	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
19	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00
20	Turn Left	Turn Left	0.00	0.00	0.00	1.00	0.00	0.00

Table 10. The Tanh function's prediction results in the turn-right gesture.

No	Aim	Predict	Down	Left	Right	Turn left	Turn right	Up
1	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
2	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
3	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
4	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
5	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
6	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
7	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
8	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
9	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
10	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
11	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
12	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
13	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
14	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
15	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
16	**Turn Right**	**Right**	**0.00**	**0.00**	**0.50**	**0.00**	**0.50**	**0.00**
17	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
18	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
19	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00
20	Turn Right	Turn Right	0.00	0.00	0.00	0.00	1.00	0.00

Table 11. Forecast results in the confusion matrix.

	Tanh-test					
	Up	Down	Left	Right	Turn left	Turn right
Up	13	3	1	2	1	0
Down	0	19	1	0	0	0
Left	0	0	0	0	0	0
Right	0	0	0	19	0	1
Turn left	0	0	1	0	19	0
Turn right	0	0	0	1	0	19

```
            Reference
Prediction   down left right turnleft turnright up
  down         20    0     0        0         0  0
  left          0   20     0        0         0  1
  right         0    0    20        0         0 12
  turnleft      0    0     0       20         0  0
  turnright     0    0     0        0        20  0
  up            0    0     0        0         0  7

Overall Statistics

               Accuracy : 0.8917
                 95% CI : (0.8219, 0.941)
    No Information Rate : 0.1667
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.87
 Mcnemar's Test P-Value : NA
```

Fig. 7. The SVM result.

Table 10 is the prediction result of the turn-right gesture. It can be seen that the test data of the 16th is wrong, and the accuracy rate is 95%.

Among the six kinds of gestures, up gesture is the one with the highest error rate. The forecast results are shown in Table 11. From Table 11, the predicted results of the six gestures can be found, and the total accuracy rate is 90%. As shown in Fig. 7, our method is also higher than the SVM. In this experiment, the kernel used in training and predicting was an SVM kernel radial basis. Also, we use package e1071 in R with the default value.

Through the above experiments and discussions, including the planning of the entire system architecture, the process of gesture recognition, and the comparison of methods, we finally made a set of gesture recognition based on the image and gesture control board using the Raspberry Pi.

4. CONCLUSION

In this paper, the gesture recognition system using the image and gesture sensing board combined with deep learning has been used, and the accuracy rate reaches 90%. We have collated some researches on gesture recognition in the literature discussion chapter, and put forward a dual-authentication gesture recognition framework, and implemented it. We use two different methods of gesture sensing board and image threshold to achieve dual authentication gesture feature extraction. In this paper, the obtained gesture image is processed by image gray-scale processing, size adjustment and key frame cutting, and 10 key frames are taken out and then predicted by deep learning.

Relevant experiments were carried out on the gesture recognition of the image, for example: the data distribution of different gestures, the experimental results of different activation functions in deep learning, and the prediction experiments of the final four gestures and six gestures. It is also slightly higher than the accuracy rate of the SVM classification method of 89%. This paper uses the Raspberry Pi as the central control center to implement a gesture recognition system.

In future research, we hope that the information on the gesture sensor board can also be taken into consideration, such as: coordinate value and waving time. For the method of hand feature extraction, there can be more in-depth research, to compare the different feature extraction methods, and advance the direction of improving accuracy. Therefore, we hope that more smart home appliances can be integrated and put into practical use in the industry.

ACKNOWLEDGMENTS

This research is supported by the Ministry of Science and Technology, Taiwan, with project number: MOST-106-2221-E-324-025 and MOST-106-2218-E-324-002.

REFERENCES

Adetiba, E., Olugbara, O.O. 2015. Lung cancer prediction using neural network ensemble with histogram of oriented gradient genomic features. Scientific World Journal, 1–17.

Alanwar, A., Alzantot, M., Ho, B.J., Martin, P., Srivastava, M. 2017. SeleCon: Scalable IoT device selection and control using hand gestures. Proceedings - 2017 IEEE/ACM 2nd International Conference on Internet-of-Things Design and Implementation, IoTDI 2017 (part of CPS Week), 47–58.

Aron, J. 2011. How innovative is Apple's new voice assistant, Siri? New Scientist, 212, 24.

Buzzi, M., Gennai, F., Leporini, B. 2018. How blind people can manage a remote control system: A case study. In Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST, 71–81.

Chai, T., Draxler, R.R. 2014. Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. Geoscientific Model Development, 7, 1247–1250.

Chen, J., Zhang, Z., Yao, L., Li, B., Chen, T. 2019. Face recognition using depth images base convolutional neural network. CITS 2019 - Proceeding of the 2019 International Conference on Computer, Information and Telecommunication Systems, 1–4.

Chen, P.H., Lin, C.J., Schölkopf, B. 2005. A tutorial on v-support vector machines. Applied Stochastic Models in Business and Industry, 21, 111–136.

Christidis, K., Devetsikiotis, M. 2016. Blockchains and smart contracts for the internet of things. IEEE Access, 2292–2303.

Chung, C.W., Chiu, L.Y. 2014. The application of gestures recognition based on depth information using shape classification and trajectory matching. Journal of Taipei University of Science and Technology, 46, 15–26.

Dai, X. 2007. Google. New Political Economy, 12, 433–442.

Dewi, C., Chen, R.C. 2019. Random forest and support vector machine on features selection for regression analysis. International Journal of Innovative Computing, Information and Control, 15, 2027–2037.

Doan, H.G., Nguyen V.T., Vu, H., Tran, T.H. 2016. A combination of user-guide scheme and kernel descriptor on RGB-D data for robust and realtime hand posture recognition. Engineering Applications of Artificial Intelligence, 49, 103–113.

FIBAR GROUP S.A., Fibaro, 2020. [Online]. Available: https://www.fibaro.com/en/. [Accessed: 17-Dec-2019].

Gregorio, F., González, G., Schmidt, C., Cousseau, J. 2020. Internet of things. in signals and communication technology, 217–245.

Haghi, M., Thurow, K., Stoll, R. 2017. Wearable devices in medical internet of things: Scientific research and commercially available devices. Healthcare Informatics Research, 23, 4–15.

Kalman, B.L., Kwasny, S.C. 2003. Why tanh: choosing a sigmoidal function.

Kelly, C. 2012. OK, Google, take a deep breath. New York Times.

Kotsidou, D. 2018. Gesture recognition. in gesture recognition: performance, Applications and Features.

Linux kernel, WiKi Raspberry Pi, https://en.wikipedia.org/wiki/Raspberry_Pi. [Accessed: 20-Oct-2020]

Li, H., Wu, L., Wang, H., Han, C., Quan, W., Zhao, J. 2020. Hand gesture recognition enhancement based on spatial fuzzy matching in leap motion. IEEE Transactions on Industrial Informatics, 16, 1885–1894.

Mitra, S., charya, T. 2007. Gesture recognition: A survey. IEEE transactions on systems, man and cybernetics part C: applications and reviews, 37, 11–324.

Newmarch, J. 2017. Raspberry Pi. Linux sound programming, 537–545.

R. R Development Core Team. 2011. R: A language and environment for statistical computing.

Risteska, S.B.L., Trivodaliev, K.V. 2017. A review of internet of things for smart home: challenges and solutions. Journal of Cleaner Production, 1454–1464.

Sakshi, B., Radha, D. 2019. Smart phone as a controlling device for smart home using speech recognition. 2019 International Conference on Communication and Signal Processing (ICCSP), 701–705.

Shi, L., Wang, Y., Li., J. 2010. A real time Vision-Based hand gestures recognition system, in ISICA 2010: Advances in computation and intelligence, 349–358.

Smisek, J., Jancosek, M., Pajdla, T. 2011. 3D with kinect. In proceedings of the IEEE international conference on computer vision, 1154–1160.

Verma, N.K., Salour, A. 2020. Feature extraction. In studies in systems, decision and control, 121–173.

Wang, C., Liu, Z., Chan, S.C. 2015. Superpixel-Based hand gesture recognition with kinect depth camera. IEEE Transactions on Multimedia, 17, 29–39.

Wenjun, T., Chengdong, W., Shuying, Z., Li., J. 2010. Dynamic hand gesture recognition using motion trajectories and key frames. In Proceedings - 2nd IEEE International Conference on Advanced Computer Control, ICACC 2010, 163–167.

Yu, X., Yuan, Y. 2019. Hand gesture recognition based on Faster-RCNN deep learning. Journal of Computers Department of Electronic Engineering, Fudan University, Shanghai, 14, 101–110.

Zhu, Y., Xu, G., Kriegman, D.J. 2002. A real-time approach to the spotting, representation, and recognition of hand gestures for human-computer interaction. Computer Vision and Image Understanding, 85, 189–208.