CONTROL HOME DEVICES BY HAND GESTURES

Hai Ninh Nguyen Thi[1], Anh Dang Thi Ngoc[1]

[1]Thai Nguyen University of Technology, Thai Nguyen city, Viet Nam

ABSTRACT

This article presents a solution to design a smart home device controller through gestures. It uses Python software and embedded in Raspberry Pi 4. The system allows users to control home appliances by hand gestures. Through the testing process, the system has met the basic requirements of the problem and can control home appliances. The system has been tested and installed at the office of the Department of Electronic Engineering - Faculty of Electronics – Thai Nguyen University of technology to demonstrate the applicability of the product.

Keyword: *Smart home, Smart controllers, gesture controllers, computer vision*

1. INTRODUCTION

Nowaday, with the explosion of the industry 4.0 technology, a series of smart devices are gradually introduced to users such as SmartPhone, SmartTV, etc. This has created a premise for Smart Home to develop strongly. Smart home is equipped with advanced automatic system for controlling lights, temperature, security, curtains, doors and many other features aimed at making life more and more comfortable, safe and secure. The smart home is equipped with a control network system and all changes and automatic control in the house are handled uniformly through the network and central processing system.

Based on the IoT, there are many different approaches in designing smart home models. Some ways can be mentioned is using mini computers such as Raspberry Pi, Orange Pi One, Pic, or Arduino, combined with online servers, MQTT servers, etc. With its compact design, used for electronic systems, setting up computing systems, DIY projects, IoT... with low cost and large open source system, Raspberry Pi has become a great choice for IoT projects, especially smart homes.
In this article, the authors focus on researching and designing a controller for smart home devices through gestures using Raspberry Pi 4. The article is divided into 5 parts: (1) Introduction; (2) Gesture control; (3) Gesture recognition based on machine vision; (4) Design hardware and software ; (5) Result; (6) Conclusion.

2. GESTURE CONTROL

A gesture can be defined as a physical movement of the hand, arm, face and body with the aim of conveying information. Therefore, gesture recognition includes not only tracking human movement, but also interpreting that movement in terms of semantically meaningful commands. From the perspective of collecting human postural information, human posture recognition technology can be divided into two categories: contact recognition technology and contactless recognition technology.
Contact recognition technology uses sensors (mechanical or optical) attached to a glove to convert finger flexions into electrical signals to determine hand posture. Although highly effective, this approach forces the user to carry a large amount of cables connected to the computer, resulting in inconvenience to the user. On the other hand, the cost of this method is also quite high.
Contactless recognition technology use computer vision, its aim is to output the structural parameters of the whole or part of a person's extremities, such as contours of the human body, position and orientation of the head, position of human joints. We see that the machine vision gesture recognition solution overcomes the limitations of Contact recognition technology, suitable for a wide range of audiences, from advanced researchers to newers. And this is also the solution that the article focuses on research. We have the block diagram of the system to control smart home devices with gestures as shown in Figure 1

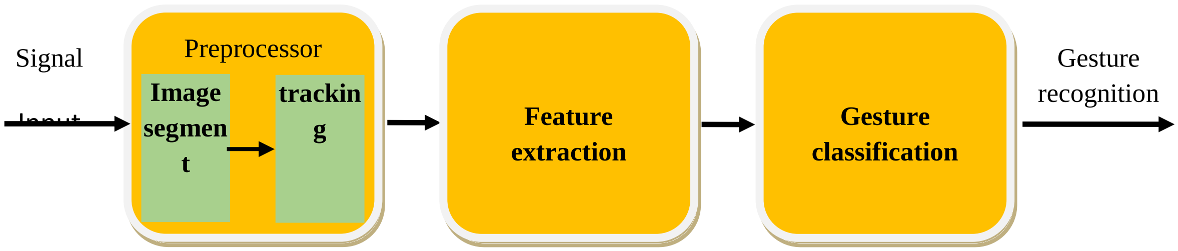


Fig -1 : The block diagram of the system

- Preprocessor block: Record images or videos with tools such as cameras. Once the object is detected, regions of interest are segmented and isolated from the image background. The tracking layer is then responsible for localizing the hands or bodies in the image sequence by performing temporal data association in consecutive image frames.

- Feature extraction block: Its function is to classify the features of the image into low-level features and high-level features to make the identification process easy.

- Gesture classification block: Based on the classification criteria, the block will classify the input gestures into appropriate categories.

3. GESTURE RECOGNITION BASED ON COMPUTER VISION

There are many different gesture recognition methods such as: color-based recognition, motion-based recognition, deep learning-based recognition and skeleton simulation-based recognition [1], [2], [3], [4], [5], [6]. In this paper, the authors only focus on the device control method based on the skeleton simulation using the Mediapipe library [2].

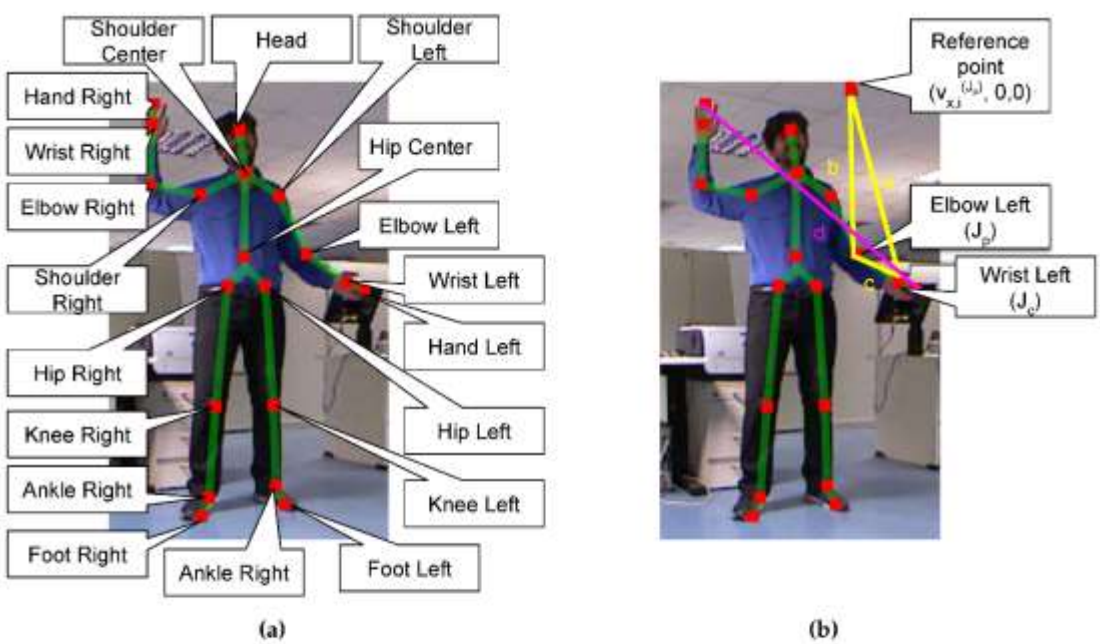


Fig -2: Skeleton identification points

For example, we can define an arm gesture as the transfer of a (sub)group of arm joints (i.e. hand, wrist, elbow) from point A to point B, using a defined trajectory relative predetermined There are many ways to extract skeleton data, however using mediapipe library is one of the simplest and most efficient methods.

4. DESIGN HARDWARE AND SOFTWARE

4.1 Design hardware

The control system includes the following main parts: Raspberry Pi 4 microcontroller block, Camera with hardware structure diagram is shown as figure 7.

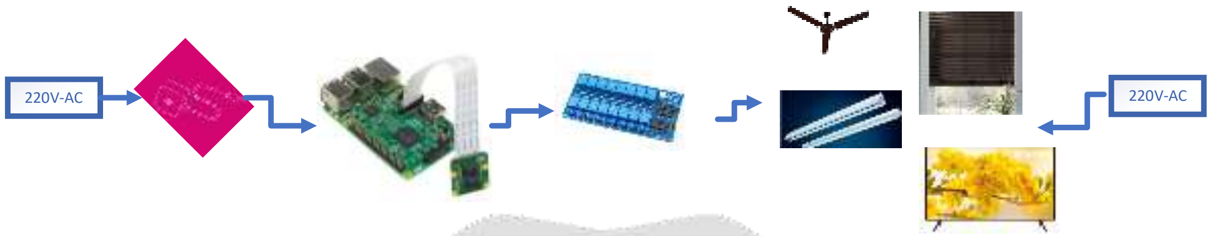


Fig -3: Hardware diagram of the system

The image obtained from the Pi camera will be processed by the Raspberry pi, based on the installed program, giving appropriate control signals to the relays to control the devices. The relay pins will be connected to the GPIO pins of the Raspberry pi as shown in Figure 10.

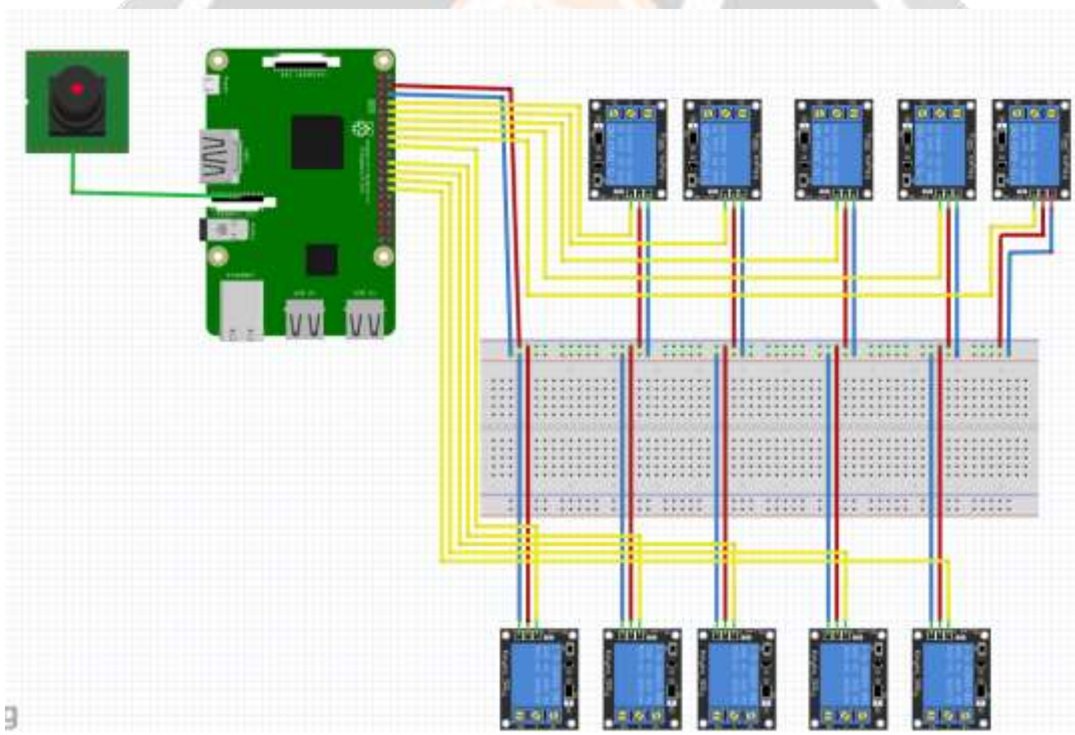
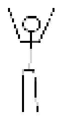
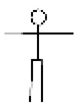
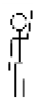
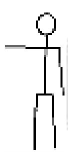
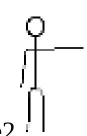
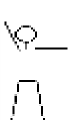


Fig -4: Hardware connection diagram

4.2 Design software

To conduct software design for the system, the authors used Python software. It is a simple, yet powerful programming language, and is equipped with features that are well-suited for processing linguistic data. When detecting a person in the frame, immediately cut the part of the frame containing the person to draw the landmark, then the tracking program will continuously monitor the object, when the object is no longer in the frame, it will call the person detection program to ensure continuity.

The gesture recognition process will be determined by the angles created by the joints, Table 1 shows the gestures corresponding control commands.

Cử chỉ	**Lệnh**	**Yêu cầu về góc**
V pose 	Control command number 1	Left elbow angle=Right elbow angle=180^0 Left shoulder angle, Right shoulder angle > 160^0
T pose 	Control command number 2	Left elbow angle=Right elbow angle=180^0 Left shoulder angle= Right shoulder angle=180^0
L pose1 	Control command number 3	Left elbow angle=Right elbow angle=180^0 Left shoulder angle=0, Right shoulder angle= 180^0
T pose1 	Control command number 4	Left elbow angle=Right elbow angle=180^0 Left shoulder angle=0, Right shoulder angle=90^0
L pose2 	Control command number 5	Left elbow angle=Right elbow angle=180^0 Left shoulder angle=0, Right shoulder angle= 180^0
T  pose2	Control command number 6	Left elbow angle=Right elbow angle=180^0 Left shoulder angle=90^0, Right shoulder angle=0
TV pose 	Control command number 7	Left elbow angle=Right elbow angle=180^0 60^0<Left Shoulder Angle<180^0, Right Shoulder Angle=90^0

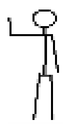
U pose2 	Control command number 8	Left elbow angle= 180^0, Right elbow angle=90^0 Left shoulder angle = 0, Right shoulder angle = 90^0

Table -1: Angle requirements for each gesture

The main tasks that need to be performed are: detecting and tracking people in the frame, drawing landmarks, calculating angles between joints and classifying gestures.

The program will have 3 main functions that perform the above functions:

- Posedetection function: This function has the function of detecting and tracking people in the frame, after detecting people in the frame, it will have to change the image from BGR format to RGB because OpenCV reads the image in BGR format (instead of BGR format). because of RGB).

After performing pose detection, you will receive a list of 33 landmarks representing the body joints of the person featured in the frame. And the program will join them together, forming a model that simulates the shape of the body.

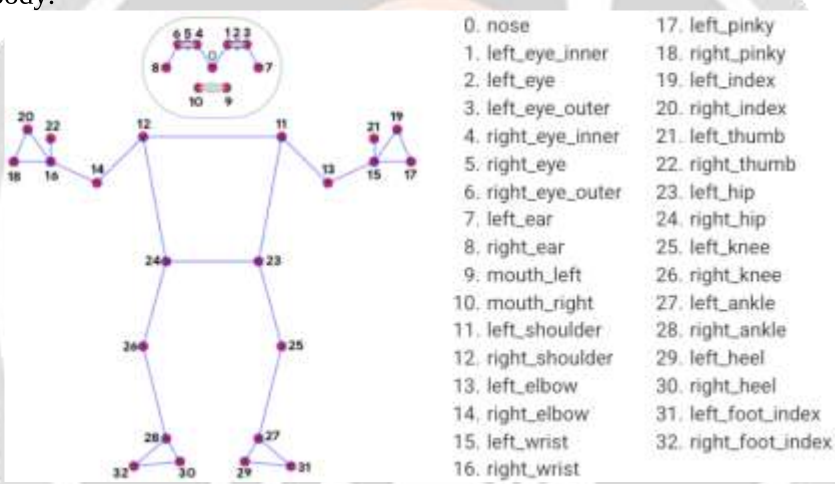


Fig -5: 33 landmarks

- Calculate Angle function: Calculates the angle formed by the joints. An angle is created by 3 independent points, so this function will calculate the angle created with any 3 consecutive waypoints out of 33 found waypoints

- The classifyPose function: This function has a gesture classification function. This function gives 6 angles to be calculated:

  - Right elbow angle: created by 3 landmarks in turn: right wrist, right elbow and right shoulder
  - Left elbow angle: created by 3 landmarks in turn: left wrist, left elbow and left shoulder
  - Right shoulder angle: created by 3 landmarks in turn: right elbow, right shoulder and right hip
  - Left shoulder angle: created by 3 landmarks in turn: left elbow, left shoulder and left hip
  - Right knee angle: created by 3 landmarks in turn: right hip, right knee and right ankle
  - Left knee angle: created by 3 landmarks in turn: left hip, left knee and left ankle

4. RESULT

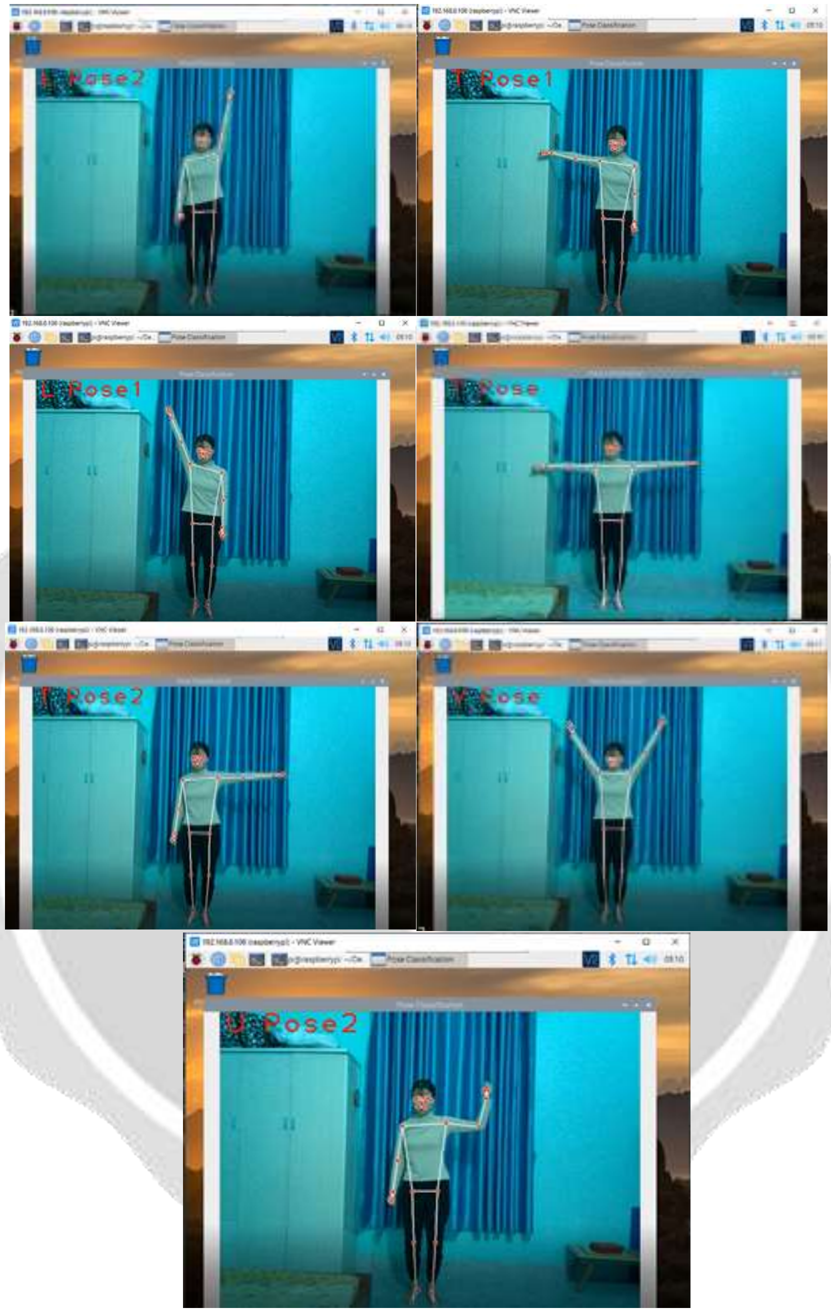


Fig -6: Gestures recognized

Figure 6 shows that the identification results are quite accurate according to the preset requirements, 33 landmarks are drawn accurately and completely.

5. CONCLUSIONS

The design solution of device controller through gestures has met the minimum requirements of a smart home. With advantages such as: simplicity, fast processing speed, no high processing requirements, fairly accurate gesture recognition, multi-gesture recognition. The system has been experimentally applied at the Department of Electronic Engineering - Faculty of Electronics - Thai Nguyen University of Industrial Technology for students to study and practice. In the coming time, the authors will make further improvements. Plus the ability to recognize gestures, reduce latency in the image processing process of Raspberry Pi.

6. ACKNOWLEDGEMENT

The authors thanks for the financial support of Thai Nguyen University of Technology for the research team to design, implement and complete the product.

6. REFERENCES

[1]. J. A. P. J. Javier Molina, "Real-time Motion-based Hand Gestures Recognition from," Journal of Signal Processing Systems. Reference 1
[2]. H. W. a. J.-P. V. Q. De Smedt, "Heterogeneous hand gesture recognition using 3D dynamic skeletal data," Computer Vision and Image Understanding.
[3]. Z. Z. a. Z. X. Wang, "A human body based on sift-neural network algorithm attitude recognition method," Journal of Medical Imaging and Health Informatics.
[4]. A. A.-N. a. J. C. Munir Oudah, "Hand Gesture Recognition Based on Computer," Journal of imaging.
[5] A. C. Calvin Ng, "Training of a deep learning algorithm for quadcopter gesture recognition," International Journal of Advanced Trends in Computer Science and Engineering.
[6] N. M. Alnaim, "Hand Gesture Recognition using Deep Learning Neural Networks".
[7] Datasheetspdf. Raspberry pi 4, Data sheet, 2019