Using Deep Learning Technology to Realize the Automatic Control Program of Robot Arm Based on Hand Gesture Recognition

Shang-Liang Chen*, Li-Wu Huang

Institute of Manufacturing Information and Systems, National Cheng Kung University, Tainan, Taiwan

Received 19 March 2021; received in revised form 17 June 2021; accepted 18 June 2021

DOI: https://doi.org/10.46604/ijeti.2021.7342

Abstract

In this study, the robot arm control, computer vision, and deep learning technologies are combined to realize an automatic control program. There are three functional modules in this program, i.e., the hand gesture recognition module, the robot arm control module, and the communication module. The hand gesture recognition module records the user's hand gesture images to recognize the gestures' features using the YOLOv4 algorithm. The recognition results are transmitted to the robot arm control module by the communication module. Finally, the received hand gesture commands are analyzed and executed by the robot arm control module. With the proposed program, engineers can interact with the robot arm through hand gestures, teach the robot arm to record the trajectory by simple hand movements, and call different scripts to satisfy robot motion requirements in the actual production environment.

Keywords: deep learning, hand gesture recognition, human robot interaction, YOLO

1. Introduction

Human-robot interaction (HRI) is mostly achieved in current manufacturing environments through teach pendants, buttons, or graphical user interfaces, and can only be controlled by a professionally trained operator. The process is tedious and time-consuming, and requires the operator to have relevant technical knowledge. For robot arm applications, however, the simplicity of control will affect the benefit significantly. In the future trend, HRI is developing toward the "person-centered" composite control mode, with the expectation that humans and robots can interact more intuitively [1]. Although it is widespread to use voice commands to control robots in the home environment, there are still limitations in using voice commands to control robot arms, computer numerical control (CNC), and other machines in the manufacturing environment. For example, in noisy industrial environments, it is difficult for even machine operators to communicate orally. It is less feasible for such unclear voice messages to be recognized by robots. Even when environmental noise is excluded, voice commands are suitable only for controlling call-and-come services and not for continuous remote control. Therefore, researchers have pointed out that it would be better to use hand gestures to give instructions in the continuous control of robots [2]. Many relevant studies are also being carried out.

Most of the hand gesture recognition methods adopted in research can be classified into two categories. The first is to collect the parameters of hand movement by wearable devices. For example, Rahim et al. [3] and Mardiyanto et al. [4] used accelerometers and gyroscopes for gesture recognition to realize human-machine interaction (HMI). The second method is based on computer vision. For example, Sarker et al. [5] realized the information collection of hand movements with Microsoft

* Corresponding author. E-mail address: slchen@mail.ncku.edu.tw
Tel.: +886-6-2757575 ext. 34221

Kinect. Due to the progress of computer vision technology, the identification speed and accuracy of object detection technology have been greatly improved, making the hand gesture recognition method based on computer vision more and more popular. Therefore, this research also adopts computer vision to realize hand gesture recognition. To reduce the difficulty of introducing the program into the actual production environment, the cost and benefit in the development cycles are also considered. The researchers decide to realize the goal of gesture recognition by a general webcam instead of using an RGBD camera such as the Kinect.

In traditional computer vision object detection, it is necessary to judge the object features by human beings and design appropriate image data pre-processing and feature extraction methods to realize the identification. The feature extraction processes designed by humans are heavily relied on these methods. Each process can only be applied to specific environments and does not have good generalization ability and robustness. Moreover, the detection speed and accuracy are not sufficient to be applied in robots in most cases. With the rapid development of deep learning network technology, traditional image recognition methods have been gradually replaced in recent years. The researchers only need to input a large number of pre-processed sample images into convolutional neural network (CNN), then the feature extraction can be completed by CNN itself. The manual design process is no longer required in selecting features, and the generalization ability of the extracted features is also better.

The technique “you only look once (YOLO)” [6-7] is used to perform real-time object detection with artificial intelligence technology. It can be applied to specific object detection [8-9], traffic flow calculation [10-11], self-driving applications [12-13], product defect detection [14-15], medical image analysis [16], etc. Creating a one-stage detection and classification process is the main developmental purpose of YOLO, which is different from most traditional detection networks. The whole image is taken as the input of the deep learning neural network. The regression on the positions and categories of bounding boxes is directly implemented at the output layer. Because only one CNN architecture is needed to determine the object position and category in the graph, the computing speed of object recognition can be improved. Since the first version released in 2015, YOLO has been widely used by developers and regarded as an important milestone for computer vision technology. The improved version of YOLOv4, released this year, is more powerful and tested by the Microsoft common objects in context (MS COCO) dataset. The result shows that the object recognition speed and accuracy are increased obviously, and the advantages over other object detection algorithms are considerable. As shown in Fig. 1 [7], YOLOv4 is twice as fast as EfficientDet and has a similar performance. Compared with the third generation, YOLOv4 has improved average precision (AP) and frame per second (FPS) by 10% and 12%, respectively.

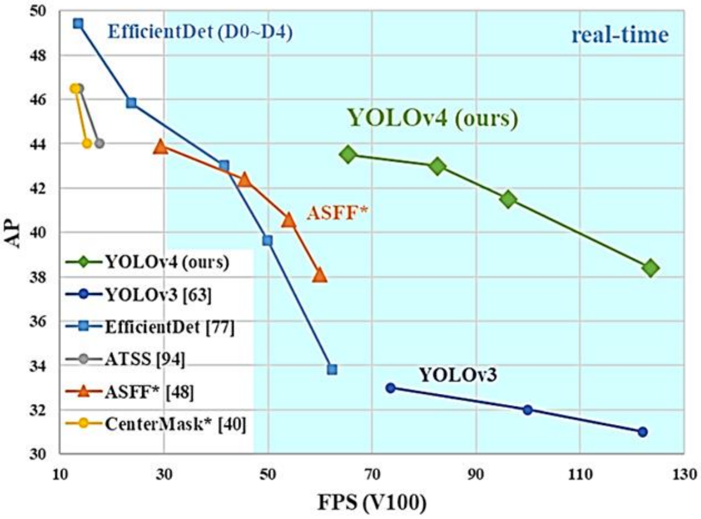


Fig. 1 The comparison of YOLOv4 and other recent object detection technology [7]

The main purpose of this study is to propose an automatic control program of robot arms based on hand gesture recognition, which makes the interaction between human and robot arms more effective. The program has the characteristics of an intuitive and convenient operation process, which can reduce the complexity of the control process of the robot arm and reduce the operator’s technical requirements. In this research, the popular one-stage object detection method YOLO is used to realize the gesture recognition module to achieve the recognition effect that cannot be achieved by traditional object detection technology.

2. Literature Review

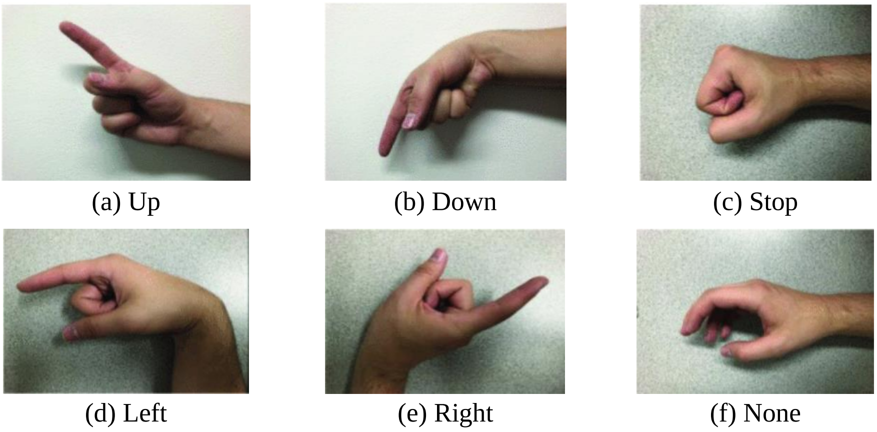
With the development of gesture recognition related technology, the related applications and research are increasing rapidly. Yasen et al. [17] collected and discussed the research related to gesture recognition published in IEEE Explore during 2016 and 2018, and 11 of them were related to robots. In the research, gesture recognition is mostly achieved by wearing devices or depth cameras. For example, in the study of Zhu et al. [18], the wristband equipped with pressure sensors was used to obtain the pressure distribution of the wrist muscles of the experimenter, and the gesture was analyzed accordingly. The gestures identified in this experiment can be divided into three kinds: wrist gestures, single finger flexions, and Chinese number gestures. Wrist gestures involve the whole hand turning the wrist, and the Chinese number gestures seem to be more complex. In the above experimental results, the wrist gestures and single finger flexions can achieve more than 95% accuracy, as shown in Table 1. However, for the more practical Chinese number gestures, only 90% accuracy, 88% recall, and 88% precision were obtained.

Table 1 Classification accuracy for each kind of gestures [18]

Subject #	Wrist gestures (%)	Single finger flexions (%)	Chinese number (%)
1	100	99	95
2	100	99	90
3	100	100	89
4	100	92	95
5	96	89	81
6	94	97	90
7	94	100	94
8	100	99	93
9	97	88	79
10	98	91	91
Average	98	95	90

Similar gesture recognition methods are also investigated in the studies of Pinzón-Arenas et al. [19] and Chung et al. [20]. In these two studies, the forearm electromyographic (EMG) data of users was collected by the Myo Armband sensor made by Thalmic Labs and used as the input of the deep learning model to realize gesture recognition. The accuracy of the supported gestures was about 85% and 99%, respectively. Although the above research achieved good accuracy, the gesture type supported by EMG is relatively simple, which can be seen in Fig. 2. Therefore, it is not possible to identify more complex and slightly varied gestures. Besides, the hand gesture recognition devices used in these studies were complex to wear, time-consuming to wear, and uncomfortable for engineers to use. Some of the wearable devices even require repeating the complex calibration process when changing operators. Therefore, using wearable devices for gesture recognition are considered unsuitable in the application environment of this research.

In the study of gesture recognition based on image vision, Luan et al. [21] used the dynamic gesture recognition system to realize the application of the robot arm picking task. They proposed a hand gesture recognition system based on an improved long-term recurrent convolutional network (LRCN) model combined with the MobileNet (V2) and long short-term memory (LSTM). Eight gestures were identified in the study, including waving the arm in different directions and gripping and relaxing the palm. The user's image is captured with a 100fps high-speed camera. The dynamic gesture model in this research achieved a good recognition effect (about 90.62% accuracy). However, the application value of dynamic gestures in a real industrial environment is not clear. As a result, this research suggests that static gesture recognition, which can be realized by a low-cost webcam, is more suitable for HMI in this kind of application.


(a) Up (b) Down (c) Stop

(d) Left (e) Right (f) None

Fig. 2 Gestures to be recognized [19]

3. Methodology

The proposed robot arm control program consists of three functional modules: a hand gesture recognition module which is responsible for recognizing and receiving control commands from the images of the machining environment, a robot arm control module that analyzes and executes the received hand gesture commands, and a communication module which is responsible for information transmission between the other two modules. The module diagram of the proposed program is shown in Fig. 3.

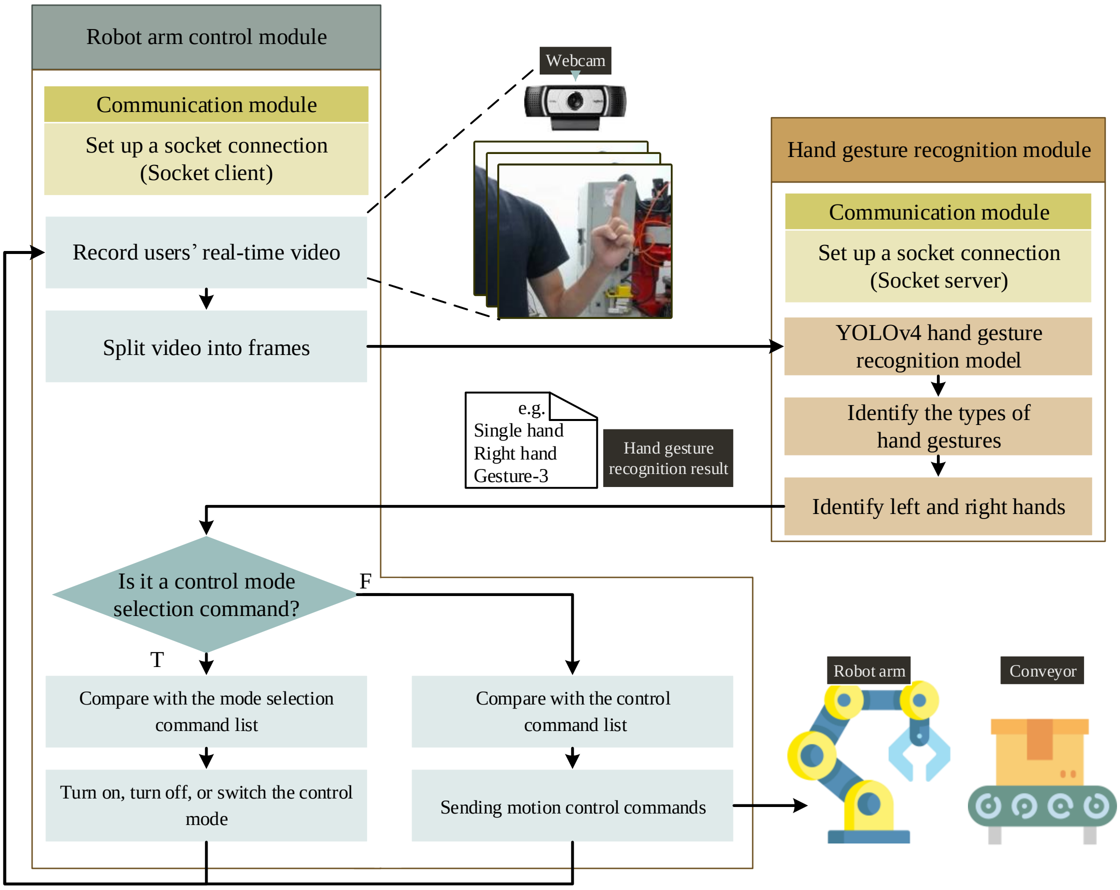
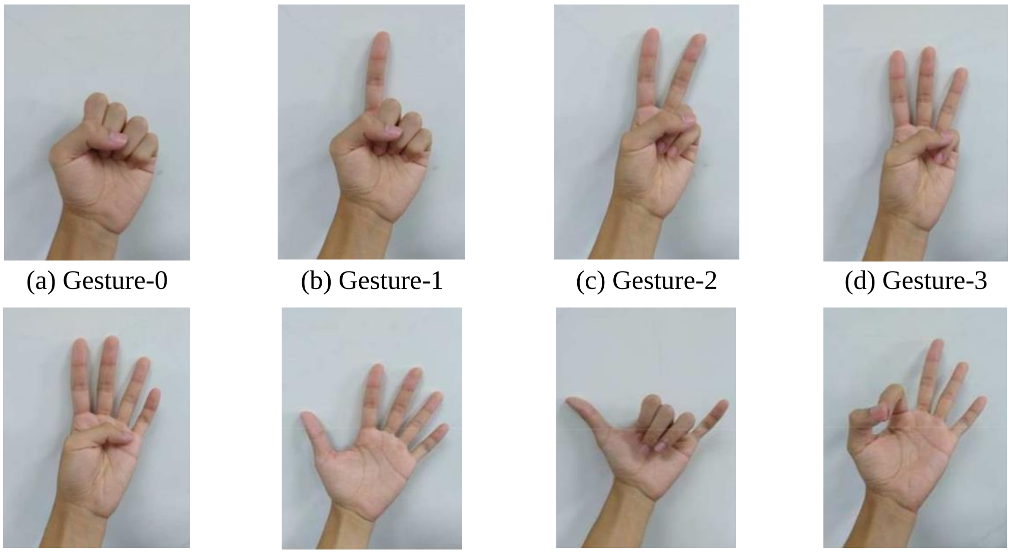

Robot arm control module
Communication module
Set up a socket connection (Socket client)
Record users' real-time video
Split video into frames
Webcam
Hand gesture recognition module
Communication module
Set up a socket connection (Socket server)
YOLOv4 hand gesture recognition model
Identify the types of hand gestures
Identify left and right hands
e.g. Single hand Right hand Gesture-3
Hand gesture recognition result
Is it a control mode selection command?
F
T
Compare with the mode selection command list
Turn on, turn off, or switch the control mode
Compare with the control command list
Sending motion control commands
Robot arm
Conveyor

Fig. 3 Module diagram of the proposed program

3.1. The hand gesture recognition module

In this study, the YOLOv4 model is trained with a customized image data set, in which Tensorflow and Keras framework are used for model training. The data set is sorted out regarding the VOC-2007 standard format proposed by Pascal organization, while the image labeling tool LabelImg is used for image labeling. Considering that a depth camera is not often used in the robot control interface in the real manufacturing environment, the hand gesture data collection in this study is only carried out through a 2D webcam.

As shown in Fig. 4, the proposed hand gesture recognition module supports a total of eight control gestures. Through this module, the robot arm operator can interact with the program by making various gesture combinations that conform to the program specifications. The program will analyze the content of hand gesture instructions, accurately control the gripper of the robot arm to move to the ideal position, adjust the rotation speed of each axis of the robot arm, trigger the conveyor of the production line, etc. Besides, to make the program practical and effective in the actual processing environment, the program also has the function of path teaching and recording so that users can store the movement process into a motion-states list. After storing all the states required by the machining process, the program can execute a complete and repeated processing program only by calling the motion-states list and setting the number of repeated execution.



(a) Gesture-0 (b) Gesture-1 (c) Gesture-2 (d) Gesture-3

(e) Gesture-4 (f) Gesture-5 (g) Gesture-6 (h) Gesture-OK

Fig. 4 Hand gestures used to interact with the control program

During gesture image collection, the researcher stands in a controlled laboratory environment and record videos about one minute and 40 seconds for each of the eight gestures designed for the program. During the process, the researcher changes the position, orientation, and distance from the camera by moving or rotating the wrist from time to time to obtain a more generalized data set. After recording all the gesture videos, the videos are converted into images at the frequency of 30 frames per second to generate an image data set of about 3,000 images for each category and a total of about 24,000 images, which are randomly assigned to test sets, validation sets, and training sets at a ratio of two to two to six. The hardware specifications used in the training model in this research are shown in Table 2. A total of 50 epochs are trained for about 23 hours.

The deep learning model after training is applied to realize the gesture recognition function of this module. This module's gesture recognition program obtains the user's real-time video stream through a webcam set up in the manufacturing environment, divides the video into frames, and inputs the frames to the object detection model. Besides, to make limited gesture commands more diversified, the program determines whether users are making gestures with both two hands, or their left and right hands, through the amount and coordinate positions of their bounding box.

Table 2 Hardware specifications

Processor	i7-9700
Generation	9th Generation Intel Core
Number of cores	8
Clock rate	3.00 GHz
Random access memory (RAM)	16 GB DDR4
Graphic processing unit (GPU)	NVIDIA GeForce GTX 1660 Ti
GPU memory	6GB GDDR6
Multiprocessor count	24
Compute unified device architecture (CUDA) cores	1536
GPU boost clock	1770 MHz

3.2. The robot arm control module

The robot arm control module in this program is responsible for analyzing the actual instruction content of the user after receiving hand gestures and sending the motion control instruction based on joint coordinates to the robot arm. Gesture instructions will be analyzed following the design of Tables 3, 4, and 5. Users select the control mode with alternating left and right hand gestures first, and then trigger the desired function according to the rules of the selected mode.

There are two control modes in the robot arm control module: direct control mode and teach control mode. In the direct control mode, the user can call various control functions in the robot arm’s motion control library through a hand gesture command to direct remote control of the robot arm and the conveyor. After the arm gripper moves to the target position, the user can record the program’s states into the state list. In the teach control mode, the user can call each state of the program recorded in the list through simple gesture instructions so that the robot arm and the conveyor can quickly execute the processing tasks taught in advance.

With the gesture instructions designed for this module, users can easily switch control modes and control the robot arm and conveyor by referring to the above list and making the correct hand gestures to the webcam set up in the manufacturing environment. The list of hand gesture instructions designed in this control module is shown in Tables 3, 4, and 5.

3.3. The communication module

Because the hand gesture recognition module of this program is implemented with Tensorflow and Keras framework, and the robot arm control module is designed with .NET framework. Therefore, to make the program operate properly, a communication module that is responsible for transmitting the commands from the hand gesture recognition module to the robot arm control module is needed to be designed. In this study, the Socket interface is used in the communication module as the data transmission pipeline. Firstly, the Socket server is established at Keras terminal, and the port number is bounded. After the program starts, the port number is continuously monitored to wait for receiving the connection demand. Then, the Socket Client is established in the .Net framework. When the user starts the program’s gesture control function, the client will transmit the connection demand to the port number monitored by the server. Then, the information exchange between the two ends can be established. After realizing the communication between the two modules, the mode of exchanging data and the corresponding actions after receiving different commands can be defined according to the program’s requirements.

Table 3 List of control instructions for mode selection state

Hand gesture command	Function
Make gesture-1 by the left hand, then make gesture-OK by the right hand.	Select and start direct control mode
Make gesture-2 by the left hand, then make gesture-OK by the right hand.	Select and start teach control mode
Make gesture-0 by the left hand, then make gesture-OK by the right hand.	Close the program

Table 4 List of control instructions for direct control mode

Hand gesture command	Function
Make gesture-1/2 by the left hand, and make gesture-1/2/3/4/5/6 by the right hand.	Jog continuously
Make gesture-1/2/3/4/5 by the left hand.	Adjust speed
Make gesture-0 by the right hand.	Pause
Make gesture-OK by the right hand.	Original return
Make gesture-0 by the left hand when the robot arm pauses.	Gripper clamp
Make gesture-5 by the left hand when the robot arm pauses.	Gripper release
Make gesture-1 and OK by the left hand when the robot arm pauses.	Conveyor turns forward
Make gesture-2 and OK by the left hand when the robot arm pauses.	Conveyor turns backward
Make gesture-3 by the left hand when the conveyor is turning.	Conveyor stops
Make gesture-6 and OK by the left hand when the robot arm pauses.	Record the status
Make gesture-6 by the right hand, then make gesture-OK by the left hand when the robot arm pauses.	Output the macro
Make gesture-0 by two hands.	Close mode

Table 5 List of control instructions for teach control mode

Hand gesture command	Function
Make gesture-0 by the right hand.	Original return
Make gesture-1/2/3/4/5/6 by the right hand.	Call state
Make gesture-OK, then make gesture-1/2/3/4/5/6 by the right hand.	Call macro
Make gesture-1/2/3/4/5 by the left hand.	Adjust speed
Make gesture-0 by two hands.	Close mode

4. Experiments and Discussion

In this study, a robot arm control program with direct control mode and teach control mode is finally realized. Users can switch between both control modes with simple two-hand gestures. The control functions include continuous jog, motion suspension, original return, motion-state record, gripper control, conveyor control, and rotation speed regulation. The function list of each control function is shown in Tables 6 and 7. Users can call all control modes through hand gesture commands and interact with the program using only one-hand or two-hand gestures.

Table 6 Functions of direct control mode

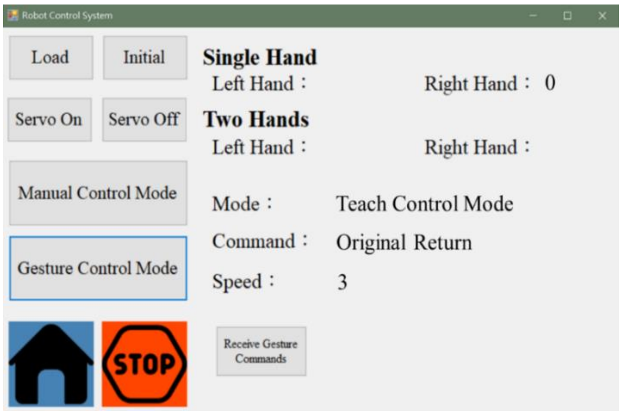
Function	Description
Jog continuously	Control one axis of the robot arm to jog continuously in a positive or negative direction.
Adjust speed	Adjust the rotation speed of all axes of the robot arm.
Pause	Pause the current motion of robot arm.
Call-and-come function	Description
Original return	Control all axes of the robot arm to return to the original point.
Gripper clamp	Control the gripper of robot arm to clamp with I/O signal.
Gripper release	Control the gripper of robot arm to release with I/O signal.
Conveyor turns forward	Control the conveyor to turn forward with I/O signal.
Conveyor turns backward	Control the conveyor to turn backward with I/O signal.
Conveyor stops	Control the conveyor to stop turning with I/O signal.
Record the status	Record the required robot arm mechanical coordinates, gripper state, and conveyor state to the motion state list, which can be called in teach control mode.
Output the macro	Output the states recorded in the state list into a motion command macro file, which can be called in teach control mode.
Close mode	Close the direct control mode and return to the mode selection state.

Table 7 Functions of teach control mode

Function	Description
Adjust speed	Adjust the rotation speed of all axes of the robot arm.
Call-and-come function	Description
Call state	Call the corresponding program state recorded in the state list.
Call macro	Call the corresponding motion command macro file.
Original return	Control all axes of the robot arm to return to the original point.
Close mode	Close the teach control mode and return to the mode selection state.

Finally, an experiment of grasping and transferring aluminum objects is designed to verify the effectiveness of the control functions of the program based on gesture recognition. The experiment is divided into two steps. The first step aims to teach the program how to process grasping aluminum objects. The experimenter first turns on the direct control mode of the program and specifies the rotation axis of the robot arm with the right hand and the rotation direction with the left hand so that the robot arm can move in the expected direction. During the process, the experimenter can also adjust the speed of the robot arm or pause the current action to ensure that the robot arm can reach the ideal position. After reaching each target position in the process, the experimenter will issue the program state recording command. The current state of each axis of the robot arm, gripper, and conveyor will be added to the motion-state list then.

The second step of the experiment is to use the recorded commands to carry out the complete process of clamping aluminum objects. The experimenter first activates the teach control mode and then starts to call the program states recorded in the list of the state with a simple single-hand command. The robot arm and the conveyor will then act in a predetermined way. Finally, the gripping of aluminum objects is complete and the objects are sent to the experimenter through the conveyor. Fig. 5 shows the interface when the instruction of original return is conducted in the teach control mode, and Fig. 6 shows the gesture recognition results at this time.



Single Hand
Left Hand： Right Hand： 0

Two Hands
Left Hand： Right Hand：

Mode： Teach Control Mode
Command： Original Return
Speed： 3

Fig. 5 The program interface when the instruction of original return is conducted in the teach control mode

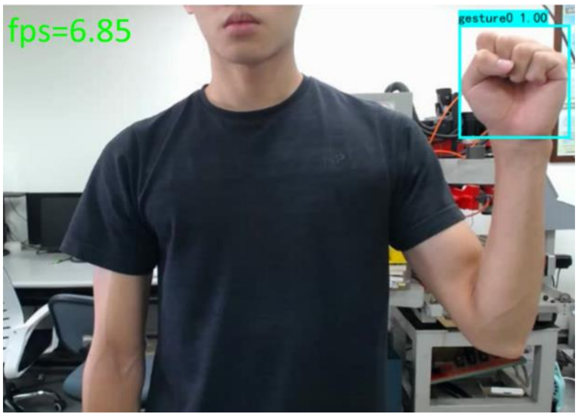
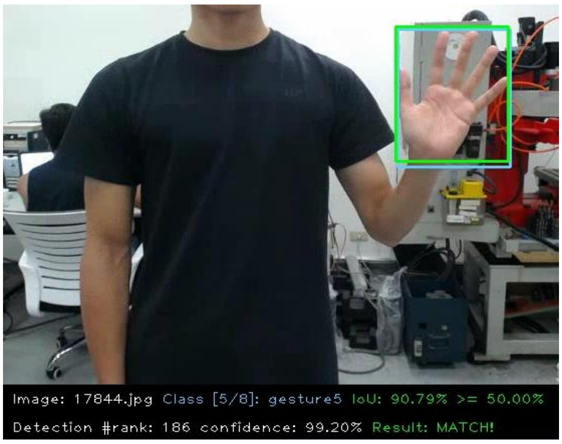


Fig. 6 The output of gesture recognition when the command of original return is given in the teach control mode

To evaluate the performance of the gesture recognition model, the mean average precision (MAP) of the model is calculated using the test set in the data set after the training. In the calculation process, the Intersection over Union (IOU) is set as 0.5. An example of the calculation results is shown in Fig. 7. The input of the model is one of the pictures in the gesture-5 test set. The green box is the position of the gesture predicted by the model, and the light blue box is the ground truth. The statistical graph of the calculated results of MAP is shown in Fig. 8.



```
Image: 17844.jpg Class [5/8]: gesture5 IoU: 90.79% >= 50.00%
Detection #rank: 186 confidence: 99.20% Result: MATCH!
```

Fig. 7 Example of MAP calculation results

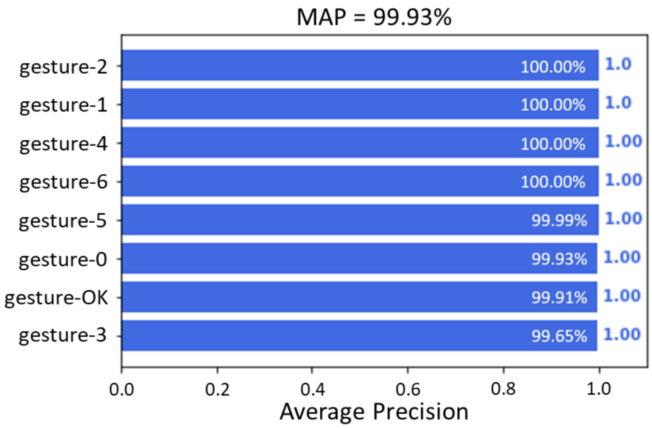


Fig. 8 Statistical graph of MAP calculation results

There is still room for improvement in the generalization ability of the trained YOLOv4 model. The above MAP is calculated using only the test set of the data set but not the gesture data of different lighting and background. Besides, because of the design of the program control function, it can only receive two-hand instructions from a single operator at present, and there is no design for the control function of multiple people and multiple gestures at the same time. Furthermore, only the pre-set specific commands can be recognized by the recognition and robot arm control modules. Users will be limited by this shortcoming and will not be able to interact freely with the program. Nevertheless, the above MAP calculation results prove that the model of the proposed robot arm control program can be introduced into the actual manufacturing environment and effectively perform hand gesture recognition. Future research will improve the above problems and optimize the overall program functions and user interface to get a perfect operation experience.

5. Conclusions

This research realized a program through visual perception function, which is different from the traditional robot arm control program, enabling the program users to control the robot arm more intuitively in the manufacturing environment. The gesture recognition function proposed in this study provides a natural human-machine interaction mode. It avoids using wearable devices such as sensor bracelets or gloves to reduce the cost and inconvenience. Through the eight gestures, users can intuitively teach the robot arm to record the trajectory, and can store the teaching results for future use.

Although the realized gesture recognition function can accurately identify the eight gestures, due to the characteristics of the recognition mode used, it cannot identify the gestures that are not among the eight gestures. If the results of this study are applied to different application environments, it is possible to expand the types of recognizable gestures according to new user requirements.

Using gestures to control a robot arm has potential applications in many fields. For example, it can be used in the medical field to prevent doctors from becoming infected while dealing with viruses, in factories to move heavy equipment, in the military field to deal with dangerous goods such as explosives, in the houses to automatically assist in cleaning and other household work, in long-term care to assist the elderly and the physically disabled, and so on. In the future, researchers can further develop the applications based on this study and extend the value of this study.

Acknowledgments

This study is financially supported by the research program of the Ministry of Science and Technology, Taiwan, R.O.C. (Project number MOST 107-2221-E-006-230-MY2) and the Industrial Technology Research Institute, Taiwan, R.O.C. (Project number B109-F007).

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] I. E. Makrini, S. A. Elprama, J. V. D. Bergh, B. Vanderborght, A. J. Knevels, C. I. C. Jewell, et al., “Working with Walt: How a Cobot Was Developed and Inserted on an Auto Assembly Line,” IEEE Robotics and Automation Magazine, vol. 25, no. 2, pp. 51-58, May 2018.

[2] Y. R. Oh, J. S. Yoon, J. H. Park, M. Kim, and H. K. Kim, “A Name Recognition Based Call-and-Come Service for Home Robots,” IEEE Transactions on Consumer Electronics, vol. 54, no. 2, pp. 247-253, July 2008.

[3] M. A. Rahim and J. Shin, “Hand Movement Activity-Based Character Input System on a Virtual Keyboard,” Electronics, vol. 9, no. 5, 774, May 2020.

[4] R. Mardiyanto, M. F. R. Utomo, D. Purwanto, and H. Suryoatmojo, “Development of Hand Gesture Recognition Sensor Based on Accelerometer and Gyroscope for Controlling Arm of Underwater Remotely Operated Robot,” International Seminar on Intelligent Technology and Its Applications, August 2017, pp. 329-333.

[5] P. P. Sarker, F. Abedin, and F. N. Shimim, “R3Arm: Gesture Controlled Robotic Arm for Remote Rescue Operation,” IEEE Region 10 Humanitarian Technology Conference, December 2017, pp. 428-431.

[6] R. Kaluri and P. R. CH, “Optimized Feature Extraction for Precise Sign Gesture Recognition Using Self-Improved Genetic Algorithm,” International Journal of Engineering and Technology Innovation, vol. 8, no. 1, pp. 25-37, January 2018.

[7] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” https://arxiv.org/abs/2004.10934, April 23, 2020.

[8] R. Laroca, L. A. Zanlorensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, and D. Menotti, “An Efficient and Layout-Independent Automatic License Plate Recognition System Based on the YOLO Detector,” https://arxiv.org/abs/1909.01754, September 04, 2019.

[9] L. Aziz, S. B. H. Salam, and S. Ayub, “Exploring Deep Learning-Based Architecture, Strategies, Applications and Current Trends in Generic Object Detection: A Comprehensive Review,” IEEE Access, vol. 8, pp. 170461-170495, September 2020.

[10] M. A. Al-qaness, A. A. Abbasi, H. Fan, R. A. Ibrahim, S. H. Alsamhi, and A. Hawbani, “An Improved YOLO-Based Road Traffic Monitoring System,” Computing, vol. 103, no. 2, pp. 211-230, February 2021.

[11] K. Khazukov, V. Shepelev, T. Karpeta, S. Shabiev, I. Slobodin, I. Charbadze, et al., “Real-Time Monitoring of Traffic Parameters,” Journal of Big Data, vol. 7, no. 1, pp. 1-20, October 2020.

[12] N. Zaghari, M. Fathy, S. M. Jameii, M. Sabokrou, and M. Shahverdy, “Improving the Learning of Self-Driving Vehicles Based on Real Driving Behavior Using Deep Neural Network Techniques,” The Journal of Supercomputing, vol. 77, no. 4, pp. 3752-3794, August 2020.

[13] F. Fernandez, A. Sanchez, J. F. Velez, and B. Moreno, “Associated Reality: A Cognitive Human-Machine Layer for Autonomous Driving,” Robotics and Autonomous Systems, vol. 133, 103624, November 2020.

[14] D. He, K. Xu, and P. Zhou, “Defect Detection of Hot Rolled Steels with a New Object Detection Framework Called Classification Priority Network,” Computers and Industrial Engineering, vol. 128, pp. 290-297, February 2019.

[15] J. Jing, D. Zhuo, H. Zhang, Y. Liang, and M. Zheng, “Fabric Defect Detection Using the Improved YOLOv3 Model,” Journal of Engineered Fibers and Fabrics, in press.

[16] F. J. P. Montalbo, “A Computer-Aided Diagnosis of Brain Tumors Using a Fine-Tuned YOLO-Based Model with Transfer Learning,” KSII Transactions on Internet and Information Systems, vol. 14, no. 12, pp. 4816-4834, December 2020.

[17] M. Yasen and S. Jusoh, “A Systematic Review on Hand Gesture Recognition Techniques, Challenges and Applications,” PeerJ Computer Science, vol. 5, e218, September 2019.

[18] Y. Zhu, S. Jiang, and P. B. Shull, “Wrist-Worn Hand Gesture Recognition Based on Barometric Pressure Sensing,” IEEE 15th International Conference on Wearable and Implantable Body Sensor Networks, April 2018, pp. 181-184.

[19] J. O. Pinzón-Arenas, R. Jiménez-Moreno, and J. E. Herrera-Benavides, “Convolutional Neural Network for Hand Gesture Recognition Using 8 Different Emg Signals,” XXII Symposium on Image, Signal Processing, and Artificial Vision, April 2019, pp. 1-5.

[20] E. A. Chung and M. E. Benalcázar, “Real-Time Hand Gesture Recognition Model Using Deep Learning Techniques and EMG Signals,” 27th European Signal Processing Conference, September 2019, pp. 1-5.

[21] K. Luan and T. Matsumaru, “Dynamic Hand Gesture Recognition for Robot Arm Teaching Based on Improved LRCN Model,” IEEE International Conference on Robotics and Biomimetics, December 2019, pp. 1269-1274.