Proceedings of the 5th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/15/20230825

Using end-to-end learning and PyAutoGUI to apply gesture recognition for human-computer interaction

Junhao Tang

School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai, 264200, China

202000800119@mail.sdu.edu.cn

Abstract. Contact-less human-machine interaction is becoming increasingly important due to the growing number of special environmental needs and accessibility situations. Gesture recognition has also been a hot topic in computer vision and machine learning in recent years. In this paper, a real-time computer manipulation system based on hand gesture recognition is studied and deployed. A relatively mature end-to-end target recognition model, the YOLOv5 model, is trained in this paper to achieve real-time detection and recognition of hand gestures. According to the result of the recognition, it is translated into the corresponding operation on the computer according to a set of rules, and then PyAutoGUI is used to actually control the computer. At the end of the research, the trained YOLOv5 model exhibited excellent performance and verified the feasibility and scalability of the solution. This is a good inspiration for developing a more convenient and efficient related software.

Keywords: computer vision; end-to-end learning; hand gesture recognition; YOLOv5; human-computer interaction.

1. Introduction

Hand gesture recognition is a technology that enables computers to recognize human hand movements and translate them into data that the computer can use. In the past, hand gesture recognition required special input devices, such as data gloves, but now this technology can be accomplished simply by analyzing the image signal acquired by the video input device, such as color camera. Further, hand gesture recognition has already become an important research direction in computer vision. It has been widely used in virtual reality, smart home, medical, and gaming.

Hand gesture recognition is more natural and convenient than traditional human-computer interaction, and more conform to human intuitive habits of use. It broadens the freedom of humans to use machines and makes it possible that get rid of the limitations of traditional hardware. Meanwhile, hand gesture recognition as a touchless data entry method provides a viable solution in situations where physical contact is not available or where there is a need for accessibility. For example, in healthcare, it can be a reliable way to allow disabilities to interact with electronic devices, which allowing them to live more independently. Because of the need like these, the speed (usually need to run in real time), accuracy (not to get the exact opposite of what the user expects), and reliability (the process is always working properly) of hand gesture recognition are very important.

DOI: 10.54254/2755-2721/15/20230825

In recent years, with the development of deep learning technology, gesture recognition methods based on deep learning have become popular of research. Deep learning models can learn features automatically and have high recognition accuracy and robustness. Although some of the existing studies have proposed and validated many hand gesture recognition solutions, most of the studies have just used traditional machine learning method [1] and dedicated sensors [2] to recognize gesture for computer control [3,4,5]. Or some of the solutions that use end-to-end learning only do the recognition part [6]. Therefore, this paper proposes a computer control and data input system with hand gesture recognition based on end-to-end learning. This paper focus on improving the speed of manipulation computers with hand gesture by simplifying the complexity of the data processing in recognition.

This paper chooses You Only Look Once v5(YOLOv5) as end-to-end learning algorithm. Compared with the previous version, YOLOv5 has improved the model structure and performance. The backbone network used in YOLOv5 is CSPDarknet [7]. Cross-Stage-Partial (CSP) structure can improve the efficiency and accuracy of the model. In addition, YOLOv5 also adopts a new method called Swish Activation, which can make the model converge faster [8]. In practical applications, YOLOv5 enables real-time target detection and achieves leading performance on some benchmark datasets, such as COCO. When YOLO recognizes hand gestures from the video image captured in the camera, the system sends the results to a computer-controlled program using PyAutoGUI. PyAutoGUI is an automation framework for python. It can be programmed to emulate input devices such as mice and keyboards or to control applications directly. PyAutoGUI is available on Windows, Mac and Linux, and supports multi-monitor control, making it possible for use in complex environments with multiple systems [9]. The combination of the two, YOLOv5 and PyAutoGUI, is the key to implementing gesture-controlled computers in this paper. In this way, real-time and efficient hand gesture recognition can be achieved, in order to control the computer with hand gestures.

2. Methods

2.1. System flowchart

Figure 1 is the complete system flowchart. The system has two parts, model training part and control part. The model training part trains the YOLOv5 model with the hand gesture dataset. When it is iterated to have a good performance, the system uses trained model for gesture recognition of images captured from the camera. The results of the recognition are passed to the control part, and the PyAutoGUI controls the computer accordingly to the different hand gestures.

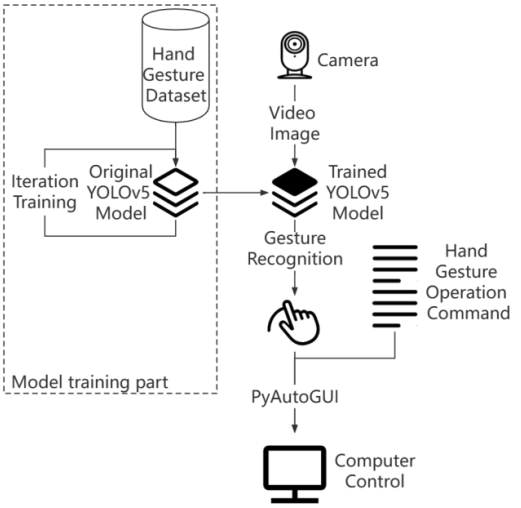

Figure 1. System flowchart, from model training to using gesture to control computer.

DOI: 10.54254/2755-2721/15/20230825

2.2. Data acquisition and pre-processing

The dataset for this study consists of two parts. The first part is the public dataset HAnd Gesture Recognition Image Dataset (HaGRID) provided by SberDevices, a Russian IT company. The dataset has a total of over 550,000 data, including 18 representative easily recognizable hand gestures, and a "no gesture" extra class [10]. All the data images are in 1920*1080 resolution. To improve training efficiency and reduce training time, the training set needs to be simplified. Here, reducing the image resolution does not affect the training results too much, and not all gestures are needed, so the simplified HaGRID dataset is used in the end. The simplified dataset has a total of about 40,000 data, which contains 6 class of hand gestures in Figure 2.

Figure 2. Selected 6 hand gesture (The names in brackets are modified by the author for subsequent use)

The second part of the data was created manually. In this study, two types of gesture data were added to improve the functionality of hand gesture manipulation. They are the leftward and rightward pointing hand gestures, like Figure 3.

Figure 3. Leftward and rightward pointing hand gestures.

2.3. End-to-end learning-based hand gesture recognition

In this paper, the YOLOv5 model developed by Ultralytics is used for training. PyTorch framework was first applied to YOLOv5 [7], making it easier and more convenient to deploy. Meanwhile, PyTorch's well-established community provides great support for the implementation of the solution.

YOLOv5 was released in multiple versions with some disparity in performance. According to YOLO publishers [11] and related studies [12,13], YOLOv5x performs the best in accuracy, but YOLOv5s improves detection speed with slightly reduced accuracy. Considering the real-time interaction as the focus of this system, it is necessary to choose the fastest model in preference. However, for the sake of research rigor, the above two versions of the model are deployed in this paper, and the actual performance metrics are given separately at the end.

DOI: 10.54254/2755-2721/15/20230825

2.4. PyAutoGUI-based computer control

PyAutoGUI is a python library for automating operations with cross-platform support for Windows, Mac, Linux and other systems. It is very flexible and can take over control of keyboard, mouse and other input devices, as well as direct control of the view window. This paper needs a simple to use but scalable way of operating to handle the various results that may be obtained from the gesture recognition process. That's why this paper has chosen PyAutoGUI. Furthermore, PyAutoGUI is an open-source library, so it is easy to modify and further apply it to various situations.

When the system is working, the gestures appearing in the camera screen are detected in real time. The result of the detection is converted into a status output, and the computer control program starts to control when it receives the change of status.

In order to control the computer with hand gestures for corresponding operations, this paper designs a rule as shown in Table 1 below.

Table 1. Hand gesture operation command.

Hand gesture	Operation
fist	0. Default status, no operation
up, down, left, right	1. Move the mouse pointer in the corresponding direction (The longer the gesture is held the faster the pointer moves)
fist→palm→fist (in 2 seconds)	2. Left mouse button click (Do the gesture twice in 2 seconds for double click)
fist→palm (Hold for more than 2 seconds)	3. Long press on the left mouse button (Dragging can be done by doing up, down, left, right gestures without returning to the fist)
fist→two up→fist	4. Right mouse button click
fist→ok→fist (in 2 seconds)	5. Minimize current focus window
fist→ok (Hold for more than 2 seconds)	6. Minimize all windows, return to desktop

Note that the system uses fist as the default state. The program only starts to take over computer control when the system accepts fist as the start signal. Any gesture needs to return to the fist state after it is completed. If a meaningless gesture combination is entered, the system will automatically return to the default state and wait for the fist signal.

3. Results & discussion

3.1. Experimentation platform

Windows 10 computer with 8GB video memory rtx2070s graphics card. CUDA version 11.6. The programming environment is python 3.9. The learning model is YOLOv5s and YOLOv5x. Based on the actual training, the average amount of video memory used is about 6.9GB when using the YOLOv5s model.

3.2. Hyper parameters

The training hyper parameters are set as shown in the following Table 2.

DOI: 10.54254/2755-2721/15/20230825

Table 2. Hyper parameters and values.

Hyper Parameters	Value
Initial learning rate	0.01
Final OneCycleLR learning rate	0.2
Epochs	300
Optimizer	Stochastic Gradient Descent (SGD)
SGD Momentum	0.937
Batch Size	32

In the model parameter settings, since the gestures to be trained in this paper have different left and right directions, it is necessary to turn off the left and right flipping of the data augmentation.

3.3. Training

This paper uses about 40,000 images for training and 1500 images for verification. Each epoch has about 1300 batches. Due to the performance limitations of the device used, the device can process about 3 batches per second. A complete epoch takes 7 minutes of training time. This is the data for using the YOLOv5s model. If the YOLOv5x model is used for training, the time for one epoch will increase to 50 minutes. It means that the total training time will be up to about 11 days. But YOLOv5s can be trained in less than 2 days. It is obvious that YOLOv5s has a very great advantage in terms of efficiency.

3.4. Performance indicator

In this section this paper gives the performance indicators such as mean average precision (mAP), box_loss, cls_loss and the others after the training of YOLOv5s model.

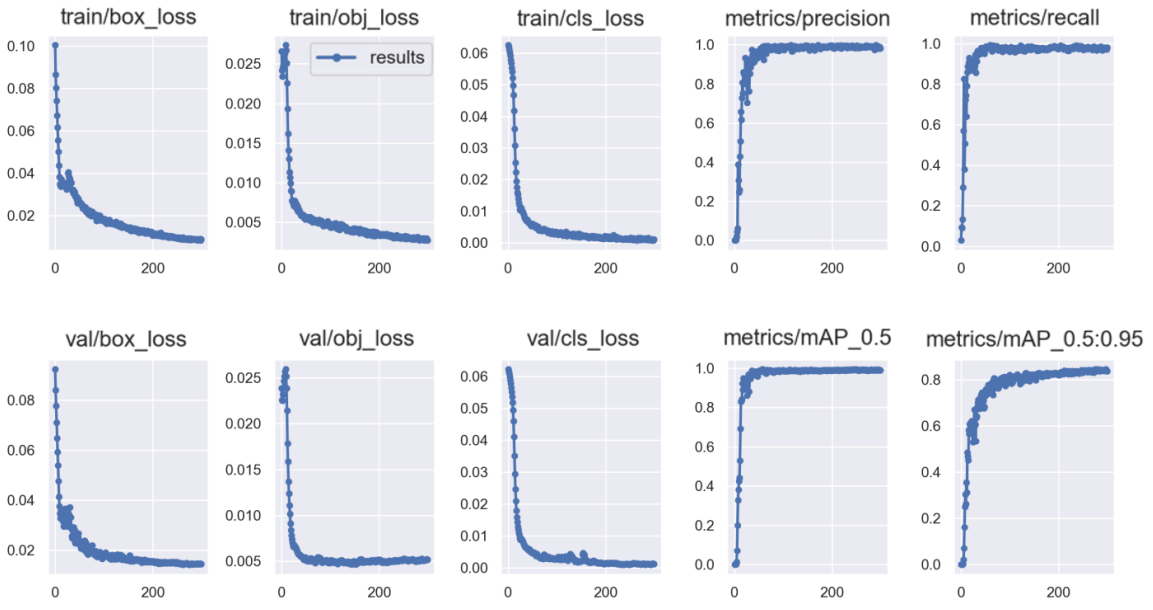

Figure 4. Loss and mAP (YOLOv5s).

According to the above scatter plot (Figure 4), the loss function drops significantly to a better metric during the previous epoch, and the accuracy converges rapidly from nearly 0% at the beginning to over 90%. After reaching a good level, the model is continuously optimized at a

DOI: 10.54254/2755-2721/15/20230825

smaller rate. Finally, an mAP@0.5 of 0.992 and mAP@0.5:0.95 of 0.838 were achieved. The following Figure 5 is PR curve.

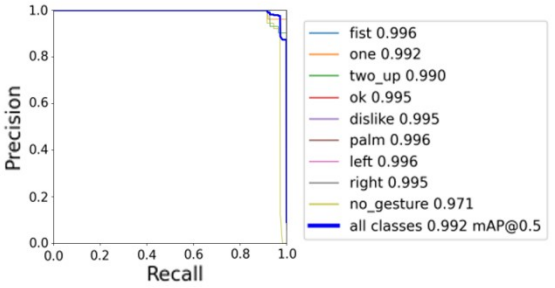

Figure 5. PR Curve (YOLOv5s).

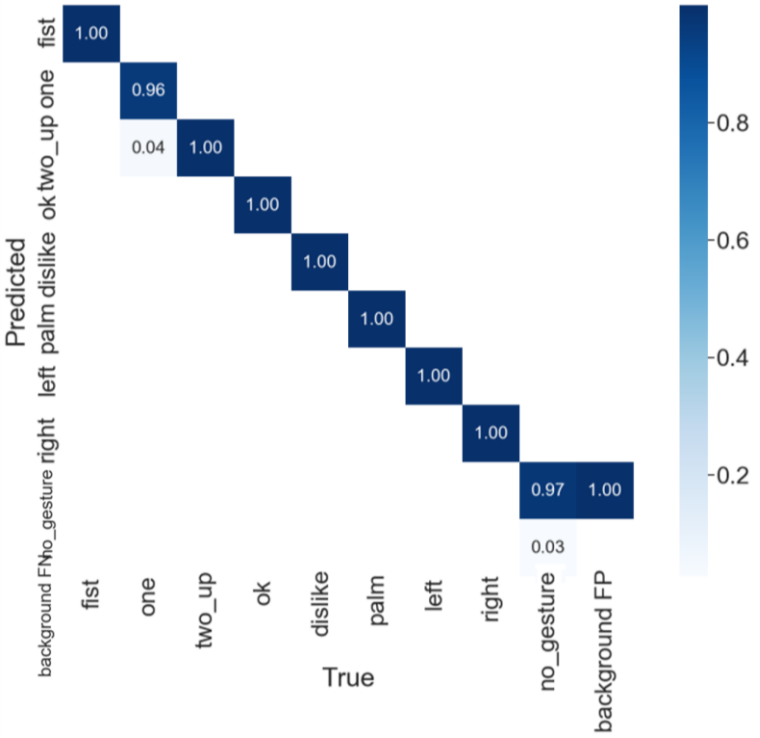

Figure 6. Confusion Matrix (YOLOv5s)s.

From Figure 6, the final trained model has good accuracy and differentiation for each different class of recognition. Most of the hand gesture categories achieved almost 100% correct results in the test. One gesture and Two_up gesture are similar in form, so there is a small probability of incorrect recognition in the test data where some of the feature information is relatively vague. Overall, the performance is still very good.

3.5. Computer control

After obtaining the trained model, the system was tested by 10 testers in this research. The system shows good recognition rate for different positions appearing in any position in the camera under various environments. The recognition speed of the system can reach 50 frames per second (limited by the

DOI: 10.54254/2755-2721/15/20230825

performance of the test computer), and the response speed meets the use demand of real-time operation. Examples of the testing process are displayed in Figure 7

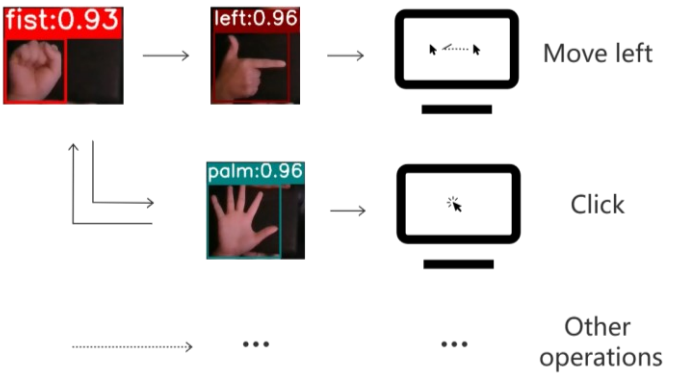

Figure 7. Recognizing hand gestures to control computer.

3.6. Discussion

What can be acknowledged is that YOLOv5s performance is very good. Because of its small model size and fast computing speed, it can complete the training in a short time that would take an extremely long time for traditional machine learning. Even compared to its predecessor, YOLOv3, YOLOv5s shows better accuracy and speed. Thanks to the end-to-end detection method, the complex processing of the data is removed, making real-time monitoring in a high frame rate environment possible. The author also tried testing with a small data set of less than 100 images, and after a very short training time, YOLOv5s also achieved good recognition of brand-new features. This means that the system allows users to enter some photos of the gestures they need to recognize, and then the system can accurately recognize these custom gestures. This has a high practical value.

However, it is a caution that hand gestures with high similarity may lead to confusion in the detection results under some conditions. The use of more different hand gestures would be beneficial to improve the recognition accuracy of this system.

In general, the performance of this system for computer control is quite satisfactory. Attributed to PyAutoGUI's open source, during the study, the modifications to the operation details were easy and reliable, and there were basically no errors reported in the use of PyAutoGUI. Moreover, this function library runs quickly and matches well the real-time gesture recognition output. It is enough to prove that the choice of PyAutoGUI as the interface to the computer controller is very appropriate.

4. Conclusion

In order to achieve more efficient and accurate computer manipulation and data entry with hand gestures, this paper constructs a computer control solution with end-to-end learning-based hand gesture recognition. Firstly, the video images containing the user's hand gestures is captured by the camera. Then after simple pre-processing, the YOLOv5 model based on end-to-end learning is used to analyze and recognize the user's different hand gestures. Finally, this paper designs a series of command correspondence. By calling the PyAutoGUI function library in python, the computer can make corresponding real-time responses to different hand gestures according to the recognition results. This paper uses some ready-made and collected data to make a training set and test set for model training. When testers use the trained system in a variety of simulated real-world usage situations, the system shows good respond speed and accuracy. Basically, it can be said that this system has some practical utility.

In the future, the author plans to apply the system to more situations and make adaptive modifications. The current system can only handle simple operations and can only select operations in a fixed set mapping. The author will try to enrich the interaction of the system so that it can accomplish more

DOI: 10.54254/2755-2721/15/20230825

complex and detailed manipulation of the computer. For example, set up a virtual keyboard that recognizes the location of gestures for keyboard input, or allow users to develop their own usage specifications. At the same time, the extension of the application device to smart devices such as cell phones is something that can be considered. It should be mentioned that the system still lacks feedback other than visual feedback, such as hearing feedback. For the purpose of accessibility, the system will be developed to include more multi-sensory feedback to help people with disabilities to use electronic devices better.

References

[1] Jagnade, G., Ikar, M., Chaudhari, N., & Chaware, M. Hand Gesture-based Virtual Mouse using Open CV. 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things 820-825.

[2] Gavale, S., & Jadhav, Y. Hand Gesture Detection Using Arduino and Python for Screen Control. 2020 International Journal of Engineering Applied Sciences and Technology, 5, 271-276.

[3] Oudah, M., Al-Naji, A., & Chahl, J. Hand gesture recognition based on computer vision: a review of techniques. 2020 Journal of Imaging, 6(8), 73.

[4] AlSaedi, A. K. H., & AlAsadi, A. H. H. A new hand gestures recognition system. Indonesian Journal of Electrical Engineering and Computer Science, 18(1), 49-55.

[5] Guo, L., Lu, Z., & Yao, L. Human-machine interaction sensing technology based on hand gesture recognition: A review. 2020, IEEE Transactions on Human-Machine Systems, 51(4), 300-309.

[6] Mujahid, A., Awan, M. J., Yasin, A., Mohammed, M. A., Damaševičius, R., Maskeliūnas, R., & Abdulkareem, K. H. Real-time hand gesture recognition based on deep learning YOLOv3 model. 2021 Applied Sciences, 11(9), 4164.

[7] Thuan, D. Evolution of Yolo algorithm and Yolov5: The State-of-the-Art object detention algorithm. 2021Journal of Electrical Engineering and Computer Science,1-10.

[8] Doherty, J., Gardiner, B., Kerr, E., Siddique, N., & Manvi, S. S. Comparative Study of Activation Functions and Their Impact on the YOLOv5 Object Detection Model. 2022, Pattern

[9] Sweigart, A. PyAutoGUI documentation. Read the Docs, 25.

[10] Kapitanov, A., Makhlyarchuk, A., & Kvanchiani, K. HaGRID-HAnd Gesture Recognition Image Dataset.2022 arXiv preprint arXiv:2206.08219.

[11] Ultralytics. YOLOv5 (Version 7.0). https://github.com/ultralytics/yolov5

[12] Sozzi, M., Cantalamessa, S., Cogato, A., Kayad, A., & Marinello, F. Automatic bunch detection in white grape varieties using YOLOv3, YOLOv4, and YOLOv5 deep learning algorithms. 2022 Agronomy, 12(2), 319.

[13] Liu, K., Tang, H., He, S., Yu, Q., Xiong, Y., & Wang, N. Performance validation of YOLO variants for object detection. 2021 International Conference on bioinformatics and intelligent computing 239-243.