A Dynamic Gesture Recognition Control File Method Based on Deep Learning

Fumin Liu, Yuezhong Wu, Falong Xiao, and Qiang Liu(✉)

Hunan University of Technology, Zhuzhou 412000, Hunan, China
liuqiang@hut.edu.cn

Abstract. In order to realize the remote control of the meeting documents in progress, the traditional method uses infrared remote control or 2.4 GHz wireless remote control. However, the shortcomings of carrying and storing the remote control, the infrared itself cannot pass through obstacles or the remote control of the device from a large angle, the 2.4 GHz cost is slightly higher, etc., this article introduces the use of PyTorch model and YOLO network gesture control to facilitate this practical problem. The plan proposes to use the PyTorch model to establish a neural network, train to achieve the purpose of classifying gestures, and use the YOLO network to cooperate with the corresponding control algorithm to achieve the purpose of controlling conference documents. The experimental results show that the proposed scheme is feasible and complete to achieve the required functions.

Keywords: Deep learning · Gesture recognition · Control algorithm

1 Introduction

With the development of artificial intelligence, the United States, Japan, South Korea and other countries have formulated their own strategic plans for the development of artificial intelligence in their respective countries. my country also mentioned "intelligent manufacturing engineering" in the five major projects in "Made in China 2025". One of the two main lines in "Made in China 2025" is "the main line of digital networked intelligent manufacturing that reflects the deep integration of information technology and manufacturing technology" [1]. The formulation of these guidelines clearly defines the need to bring intelligence to all aspects of our lives. In this process, good human-computer interaction is an important foundation of "human-machine co-prosperity" and "human-machine collaboration". Among them, human-computer interaction includes three types: human-computer interaction (HCI), human-machine interaction (HMI), and human-robot interaction (HRI) [2]. This has gone through a long period of development, from people using binary codes to control computers to Today's voice or gesture control machines, scientists have gone through unremitting efforts [3].

Gesture is an important way of non-verbal communication between people and an important way of interaction between humans and machines. But before implementing

M. R. Khosravi et al. (Eds.): CloudComp 2021, LNICST 430, pp. 30–39, 2022.
https://doi.org/10.1007/978-3-030-99191-3_3

gesture control, define what a gesture is. In 1990, Eric Hulteen and Gord Kurtenbach published an article entitled Gestures in Human-Computer Interaction, which defined the question of what gestures are. The core is that gestures can be used to communicate, and the meaning of gestures lies in telling rather than executing. After clarifying the definition of gestures, gesture recognition also has different implementation methods. It is roughly divided into two categories according to whether the gesture recognition device is in contact with the body: contact gesture recognition and non-contact gesture recognition. Among them, contact gesture recognition is the most widely known. For example, the release of the first iPhone in 2007 is also praised by many as the creation of the era of smart phones. Among them, gesture taps account for a large proportion of the smart word. Non-contact gesture recognition, for example, references the Kinect3D somatosensory camera, the BMW iDrive system that introduces the gesture recognition function, etc. [4].

In the early stage of gesture development, people mainly used external auxiliary tools, such as a large number of sensors, to realize gesture recognition. Its advantage is that the recognition accuracy is high, but it needs the assistance of a large number of sensors, which is relatively cumbersome to implement and cannot be achieved by simply using a computer. Non-contact gesture recognition is mainly based on the feature extraction of gestures, using skin color, shape and other features to segment the gestures, and then using support vector machine (SVM) and other classification algorithms for recognition. The difficulty of this method is how to extract and segment the gestures from the environment. If the extraction algorithm is not perfect or the model training set is insufficient, the accuracy of gesture recognition is not high. After a long period of development, Grobel et al. used Hidden Markov Model to complete gesture recognition with an accuracy of 94%. Although the recognition effect is gratifying, it still needs to use an external sensor to obtain the tester's hand skin color to distinguish it. Popularize this method on a large scale [5]. The gesture recognition method based on Kinect depth information proposed by Dominio et al. has great accuracy and can reach 99.5% of recognition accuracy, but its algorithm is relatively complex and requires high equipment implementation [6]. The deep learning method proposed by Wang Xiaohua can better obtain the high-level and multi-dimensional features of past images, and can realize detection and recognition in complex environments. And the algorithm complexity is not high, but the detection speed is not high. Liu Zhijia et al. proposed a method that can use infrared image detection to improve the accuracy of gesture recognition, but infrared photoelectric technology is needed to detect the heat radiation signal of an object, which still has relative limitations.

After several years of initial development, non-contact gesture recognition has become the mainstream, and the direction of researchers has gradually changed to the realization and optimization of non-contact gesture recognition in different categories and in different situations. In 2014, Yu Jing et al. proposed a fast dynamic recognition algorithm based on the depth information of the Kinect sensor. The depth image was acquired through the Kinect depth camera, the depth image was preprocessed by the threshold segmentation method, and the foreground was extracted by the OpenCV function library [7]. In 2015, Zhao Feifei and others used the Kinect camera to design a set of gesture recognition algorithms to obtain bone information. The algorithm detects

the start and end points of the gesture, then intercepts and pushes to extract the gesture features, and finally uses the distance-weighted dynamic time warping algorithm to calculate the test samples, Get the recognition result [8]. At the same time, Redmon et al. proposed a detection YOLO algorithm at the 2016 CVPR conference, which greatly shortened the detection time. In 2018, Wu Xiaofeng and others proposed a gesture recognition algorithm based on Faster R-CNN, and modified the key parameters of the Faster R-CNN framework to achieve the purpose of detecting and recognizing gestures at the same time [9].

2 Construction of PyTorch Model, YOLOv4 and Realization of Control Algorithm

2.1 PyTorch Model

PyTorch is a Python-based scientific computing package, originally developed by the Facebook artificial intelligence research team, and its underlying layer is implemented in C++. PyTorch has two major features [10], one can use tensor calculations similar to Numpy, and GPU or CPU can be used to accelerate the speed of the training set [11]. And the common graphics card NVIDIA configures the corresponding CUDA for the graphics card after MX150 so that researchers can use GPU acceleration [12]. Then it is a deep neural network with an automatic differentiation system, which can quickly build a neural network and has a rich API interface for researchers to use [13].

Using PyTorch to build a neural network can be achieved by using the torch.nn package. Each nn.Module contains various layers and a forward (input) method, which returns output. The realization of a simple digital classification neural network is shown in Fig. 1.

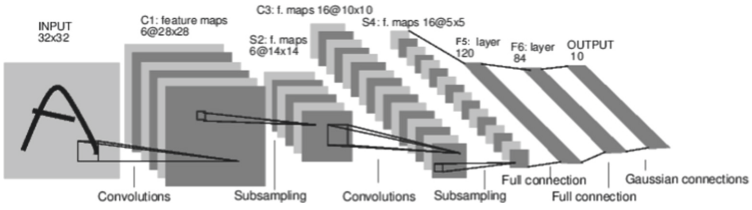

Fig. 1. Implementation diagram of digital classification neural network

On the basis of this example, the image, text, audio or video data we need to process is loaded into a numpy array using the Python standard library before building a neural network, and then the array is converted to torch.*Tensor. For pictures, Pillow or OpenCV can be used for processing [14].

2.2 Introduction to YOLO

The YOLO algorithm was published "You Only Look Once: Unified, Real-Time Object Detection" by Joseph Redmon in CVPR2016. As the name suggests, the author emphasizes the single-stage model, as shown in Fig. 2.

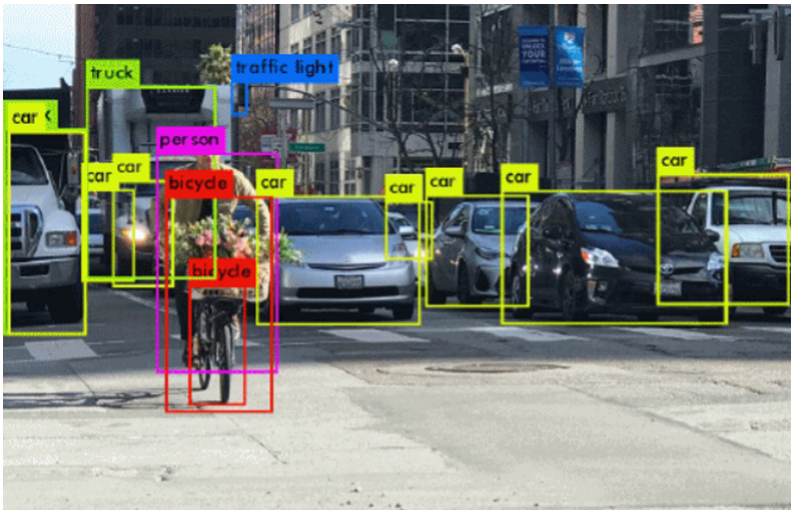

Fig. 2. Single-stage model diagram of YOLOv1 algorithm

After three iterations of YOLOv1, YOLOv2 and YOLOv3, the YOLO algorithm has matured. The structure of YOLOv1 is to add 4 convolutional layers and 2 fully connected layers on the 20 layers of the GoogleNet network [16]. Divide each image into 7 × 7 grids. When an object in the image exists in one of the grids, a single grid is responsible for judging the object. Each grid is assigned 2 bounding boxes, and finally a 7 × 7 × 30 tensor is output. The number of channels 30 is the number of channels, including 5 coordinate information of 2 bounding boxes [15]. Its confidence is shown in Eq. 1.

$$c_c = \mathrm{p}(*)U_t^p \tag{1}$$

Where P(*) indicates whether there is an object center in the grid, if it exists, it is 1, otherwise it is 0. U indicates the intersection ratio of the predicted frame and the real frame. The calculation formula is shown in formula 2.

$$U = \frac{B_T \cap B_P}{B_T \cup B_P} \tag{2}$$

The YOLOv2 algorithm is improved on the basis of YOLOv1. Using the structure of Darknet19, using a small convolution kernel operation. And YOLOv2 draws on the Faster R-CNN algorithm and uses anchor boxes to generate more detection boxes for each small table that is segmented [17]. And in the selection of the anchor point frame, the k-means clustering algorithm is used to select the anchor point frame closer to the detected object, which makes the network bracelet faster and easier to learn [18]. The calculation formula is shown in formula 3

$$\mathrm{d}(b, c) = 1 - U(b, c) \tag{3}$$

C is the cluster center; U(b,c) is the intersection ratio of the center box and the ground truth box. YOLOv2 also uses batch normalization to process the input of each layer, which greatly improves the training speed and prevents overfitting.

YOLOv3 uses a new convolutional network Darknet-53 on the basis of YOLOv2 to extract features of target detection objects, and uses multi-scale features to detect target detection objects. While ensuring the detection speed of YOLOv2, the accuracy of predicting objects is improved. The new Darknet-53 network has 53 convolutional layers in order to have more accurate small grid segmentation [19]. Correspondingly, a 1 × 1 convolution kernel is also used to reduce the number of feature channels. As shown in Fig. 3.

	Type	Filters	Size	Output
	Convolutional	32	3×3	256×256
	Convolutional	64	3×3/2	128×128
1	Convolutional	32	1×1	
	Convolutional	64	3×3	
	Resnet unit			
	Convolutional	128	3×3/2	64×64
2	Convolutional	64	1×1	
	Convolutional	128	3×3	
	Resnet unit			
	Convolutional	256	3×3/2	32×32
8	Convolutional	128	1×1	
	Convolutional	256	3×3	
	Resnet unit			
	Convolutional	512	3×3/2	16×16
8	Convolutional	256	1×1	
	Convolutional	512	3×3	
	Resnet unit			
	Convolutional	1 024	3×3/2	8×8
4	Convolutional	512	1×1	
	Convolutional	1 024	3×3	
	Resnet unit			
	Avgpool		Global	
	Connected		1 000	
	Softmax			

Fig. 3. Darknet-53 network diagram

In Fig. 3: Type is the type; Filters is the number of convolutions; Size is the size; Output is the output; Convonlutional is the convolution; Resnet unit is the Resnet residual unit; Avgpool is the average pooling; Global is the global; Connected is the full connection; Softmax is a Softmax classifier.

Compared with YOLOv3's Darknet-53, YOLOv4 uses CSPDarknet-53, and introduces the SPP-Net structure, so that the YOLOv4 network can adapt to different sizes of input. At the same time, PANet is also introduced, making full use of feature fusion. And YOLOv4 has also been improved on the input end of the training model, mainly using Mosaic, cmBN, and SAT self-adversarial training, with Mosaic being the focus [20]. Figure 3 describes the architecture of the YOLOv4 algorithm in detail. The CSP-DarkNet53 network structure, which is the most obvious improvement over YOLOv3, is shown in Fig. 4.

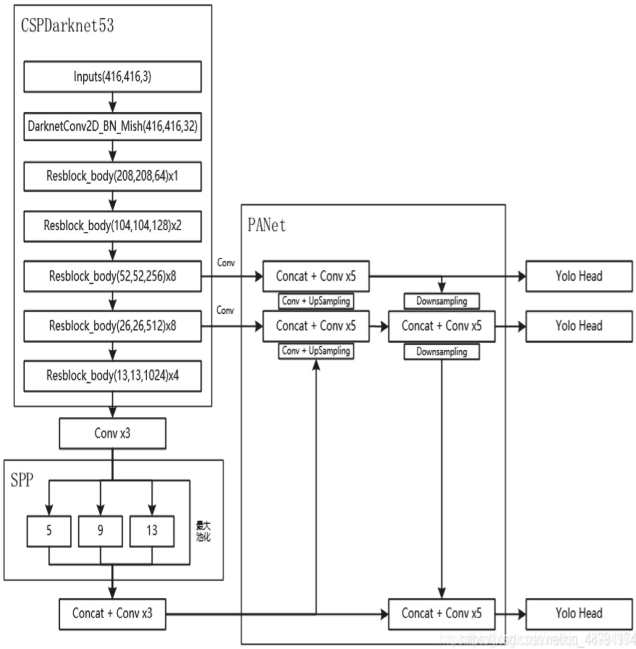

Fig. 4. CSPDarkNet53 network structure diagram

2.3 Control Algorithm

The main application scenario during the conference is the use of PPT. This article takes PPT as an example to implement the control algorithm. The main idea is to use the pyHook package in the Python library to call the global mouse and keyboard events in Windows to provide callbacks. Python applications register event handlers for user input events, such as left and right mouse buttons, and set keyboard and mouse hooks. After the model training, the gesture recognition result is judged, so that the mouse and the keyboard react correspondingly to achieve the purpose of controlling the PPT. The control algorithm is presented in pseudo code as follows:

```
1  list = []
2  while i < strlen(string)
3  do list.append(i)
4  result ← ''.join(list[2:8])
5  if result = “dianji”:
6  then k.tap_key(k.function_keys [5])
7  elseif result = “xuanzhuan”:
8  then k.tap_key(k.control_key)
```

3 Experiment and Result Analysis

3.1 Experimental Platform and Data Set

The experimental environment uses Window10 operating system and PyTroch + YOLO algorithm framework. On the hardware configuration, the CPU uses Intel(R) Core™ i7-8550U CPU @ 1.8 GHz; the GPU is the NVIDIA MX150 2G independent display.

The experiment uses an autonomous data set, which is collected from the real hand of the researcher. The image contains detection targets of different gestures at multiple scales, which is suitable as a data set for gesture detection. Part of the image is shown in Fig. 5. The data set contains 1150 images with a resolution of 320 × 240, labeled with 5 categories: xuanzhuan (rotation), pingyi (translation), zhuaqu (grabbing), suofang (zooming), and dianji (clicking).

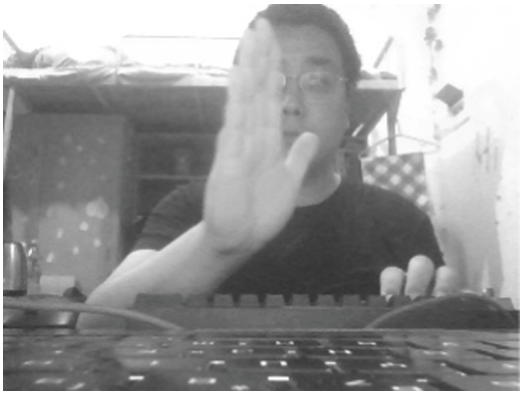

Fig. 5. Example of gestures

To achieve the effect, take the dianji (click) action as an example. First, make a click gesture as shown in Fig. 6. After the model predicts a click gesture, the control algorithm makes the corresponding action of playing PPT as shown in Fig. 7.

3.2 Network Training

The data set is divided into training set and test set at a ratio of 7:3. Use the basic YOLOv4 algorithm and PyTorch model to train the data set and get the result.

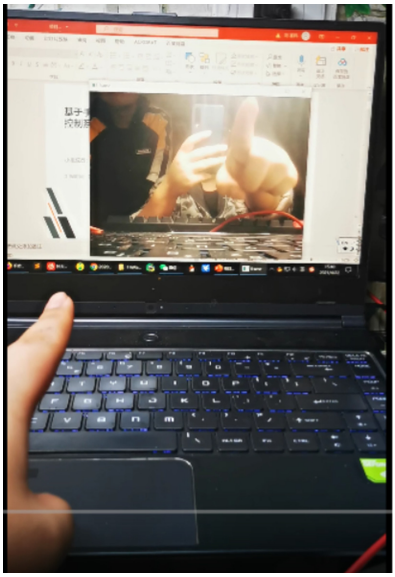

Fig. 6. An example of the "dianji" gesture

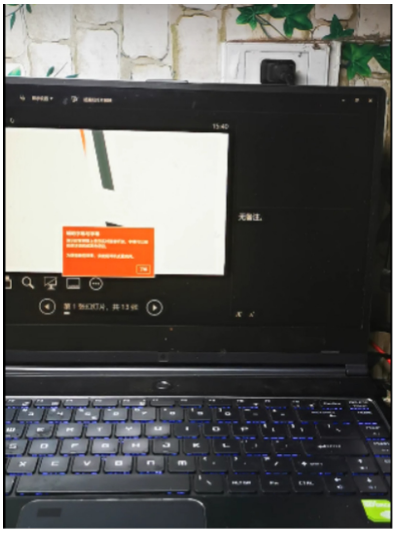

Fig. 7. "dianji" gesture realization effect diagram

3.3 Result Analysis

As shown in Table 1, this experiment uses the accuracy rate P (Precision) as the evaluation standard of the model recognition effect.

Table 1. Different gesture recognition effects

Gesture	Precision/%
PingYi	45.21
XuanZhuan	56.54
DianJi	46.45
SuoFang	42.54
ZhuaQu	59.23

The experimental results show that the method proposed in this paper has preliminary usability. The model can roughly meet the requirements. And can complete the preliminary experimental requirements.

4 Conclusion

The gesture control model using PyTorch + YOLOv4 proposed in this paper has achieved preliminary results. Through the preliminary construction of the PyTorch neural network and the application of the YOLOv4 algorithm, combined with the control algorithm through theoretical analysis and experimental verification, the following conclusions are drawn.

(1) Use PyTorch to quickly build a preliminary neural network model and get preliminary experimental data
(2) The YOLOv4 algorithm has significantly improved the accuracy and speed of YOLOv3, and the hardware requirements are not high

The next research direction can realize the gesture control from simple conference control to the field of smart home, so as to realize a richer and smarter living environment. Or, gesture control can play a role in the field of Internet of Things to realize the interconnection of everything.

Acknowledgment. This work is supported in part by National Key R&D Program Funded Project of China under grant number 2018YFB1700200 and 2019YFE0122600, in part by the Hunan Provincial Key Research and Development Project of China under grant numbers 2019GK2133, in part by the Natural Science Foundation of Hunan Province under grant number 2021JJ50050, 2021JJ50058 and 2020JJ6089, in part by the Scientific Research Project of Hunan Provincial Department of Education under grant number 19B147,in part py the Key Project of the Department of Education in Hunan Province (19A133), in part by the Degree and Postgraduate Education Reform Research Project of Hunan Province Department of Education under grant number 2020JGZD059, in part by the Teaching Reform of Ordinary Colleges and Universities Research project of Hunan Province Department of Education under grant number HNJG-2021–0710,in part by the Open PlatformInnovation Foundation of Hunan Provincial Education Department(grant no. 20K046),and this research was supported by the Special Fund Support Project for the Construction of Innovative Provinces in Hunan (2019GK4009).

References

1. Qi Jing, X., Kun, D.: Research progress of robot visual gesture interaction technology. Robot **39**(04), 565–584 (2017)
2. Bowen, S., Feng, Y.: Monocular camera dynamic gesture recognition and interaction based on deep learning. J. Harbin Univ. Sci. Technol. **26**(01), 30–38 (2021)
3. Fenhua, W., Chao, H., Bo, Z., Qiang, Z.: Gesture recognition based on YOLO algorithm. J. Beijing Inst. Technol. **40**(08), 873–879 (2020)
4. Xuanheng, L., Baosong, D., Yu, P., Bohui, F., Liang, X., Ye, Y., Erwei, Y.: Research on wearable gesture interaction system and recognition algorithm. Small Microcomput. Syst. **41**(11), 2241–2248 (2020)
5. Xiaoyan, X., Huan, Z., Lin, J.: Dynamic gesture recognition based on the characteristics of video data. J. Beijing Univ. Posts Telecommun. **43**(05), 91–97 (2020)
6. Mengli, S., Beiwei, Z., Guanghui, L.: Real-time gesture recognition method based on depth image. Comput. Eng. Des. **41**(07), 2057–2062 (2020)
7. Xiaoping, Y., Xuqi, M., Sai, L.: Improved YOLOv3 pedestrian vehicle target detection algorithm. Sci. Technol. Eng. **21**(08), 3192–3198 (2021)
8. Xiaohu, N., Lei, D.: Overview of typical target detection algorithms for deep learning. Appl. Res. Comput. **37**(S2), 15–21 (2020)
9. Zhijia, L., Xuan, W., Jinbo, Z., Yinhui, X., Xuhui, G.: Improved method of infrared image target detection based on YOLO algorithm. Laser Infrared **50**(12), 1512–1520 (2020)
10. Maurya, H., et al.: Analysis on hand gesture recognition using artificial neural network. Ethics Inf. Technol. **2**(2), 127–133 (2020)
11. Wang, Y., Yang, Y., Zhang, P.: Gesture feature extraction and recognition based on image processing. IIETA **37**(5), 873–880 (2020)
12. Kiselev, I.V., Kiselev, I.V.: Comparative analysis of libraries for computer vision OpenCV and AForge. NET for use in gesture recognition system. J. Phys. Conf. Ser. **1661**(1), 012048 (2020)
13. Liang, C.Y., Yuan, C.H., Feng, T.W.: Hand gesture recognition via image processing techniques and deep CNN. J. Intell. Fuzzy Syst. **39**(3), 4405–4418 (2020)
14. Hoshang, K., et al.: A new framework for sign language alphabet hand posture recognition using geometrical features through artificial neural network (part 1). Neural Comput. Appl. **33**(10), 4945–4963 (2020)
15. Oudah, M., Al-Naji, A., Chahl, J.: Hand gesture recognition based on computer vision: a review of techniques. J. Imaging **6**(8), 73 (2020)
16. Kai, Z., Yi, W., Hailong, H.: Research on recognition and application of hand gesture based on skin color and SVM. J. Comput. Methods Sci. Eng. **20**(1), 269–278 (2020)
17. Mo, T., Sun, P.: Research on key issues of gesture recognition for artificial intelligence. Soft. Comput. **24**(8), 5795–5803 (2020)
18. Shangchun, L., et al.: Multi-object intergroup gesture recognition combined with fusion feature and KNN algorithm. J. Intell. Fuzzy Syst. **38**(3), 2725–2735 (2020)
19. Li, X.: Human–robot interaction based on gesture and movement recognition. Signal Process. Image Commun. **81**, 115686 (2020)
20. Baldissera, F.B., Vargas, F.L.: A light implementation of a 3D convolutional network for online gesture recognition. IEEE Lat. Am. Trans. **18**(2), 319–326 (2020)