Application of Artificial Intelligence in Flower Recognition

Pingchuan Ma

Department of information and intelligent engineering, Shanghai Publishing and Printing College, Shanghai, 200093, China

teachermpc@163.com

Abstract. This paper introduces a programmable robot with AI-oriented scratch extensions by analyzing its implementation architecture and operating procedure, and those of a corresponding mini-app for wireless configuration. It explains some typical extensions and their key building blocks, such as costume and backdrop and image classification. It also illustrates the fundamental idea and steps for artificial intelligence, especially machine learning, by implementing a robot to recognize two different categories of flowers.

Keywords: artificial intelligence machine learning programmable robot scratch programming image classification raspberry pi Jetson Nano

1 Introduction

In recent years, with breakthroughs in core algorithms, rapid improvement in computing power, and the support of massive internet data, artificial intelligence has witnessed a qualitative leap and become the focus of attention of countries around the world. The landing application of artificial intelligence in the field of education is the trend [1,2]. Learning artificial intelligence is challenging because it involves profound mathematical and computer knowledge, complex algorithms and models that are difficult to understand, and many other prerequisite knowledge such as Python or TensorFlow [3]. This has become the biggest obstacle for many students to learn artificial intelligence.

Scratch [4] is a simple graphical programming tool developed by the Massachusetts Institute of Technology. It is a "block-style programming" that creates code by dragging and stacking blocks. Unlike programming languages such as Python and C, Scratch does not require writing text code, nor does it need to comply with "rigid" syntax formats. For students who are new to programming, starting with "block-style programming" is the best choice. Scratch programming has many benefits, such as cultivating logical thinking ability, improving learning initiative, and stimulating creativity. Practice has proved that almost all children will love this software at first sight.

C. Chen et al. (eds.), *Proceedings of the 3rd International Conference on Digital Economy and Computer Application (DECA 2023)*, Atlantis Highlights in Computer Sciences 17,
https://doi.org/10.2991/978-94-6463-304-7_21

This article introduces the artificial intelligence plugin of Scratch, and elaborates on the basic ideas and processes of artificial intelligence by creating a flower recognition robot.

2 Introduction to Robot Functions

Programmable robots include a camera for image capture, motors and wheels for movement and steering, a microphone and speaker for audio input and output. In addition, programmable robots also have wired and wireless networks for convenient access to the Internet and remote monitoring.

Programmable robots can be easily connected to a computer or tablet through a mobile hotspot or wireless router, provided they are on the same local area network. Programmable robots support different protocols such as HTTP, SSH, and VNC for connection.

3 Robot Implementation Architecture

The programmable robot is based on Raspberry Pi [5] or Jetson Nano [6] and uses the Ubuntu operating system [7]. It is installed with NGINX [8] and APACHE services [9]. When connecting to the programmable robot via HTTPS protocol from a computer or tablet, the Scratch front-end code is downloaded to the computer or tablet and displayed in a web browser that supports HTML5 [10].

The front-end code includes artificial intelligence plugins, which communicate with the scratchd service process running on the programmable robot through AJAX requests sent via GET or POST methods. The scratchd service process is responsible for executing tasks such as model training, motor driving, and camera driving.

To facilitate the robot's access to the IP address, a WeChat mini program has been developed, which uses Bluetooth to configure wireless access for the programmable robot.

WeChat mini programs are all front-end code and are published on the Tencent WeChat public platform after being reviewed. After logging into WeChat on a tablet or smartphone, the mini program code can be downloaded and run on the device. The mini program code includes a front-end for Bluetooth configuration. When Bluetooth is enabled on the tablet or smartphone, the programmable robot is searched, and communication is established with the programmable robot through the Bluez protocol.

Once the programmable robot is started, Bluetooth remains on. At the same time, the bluetoothd service process listens for requests from tablets or smartphones and returns device models, IP addresses, Wi-Fi information, or performs Wi-Fi configuration operations.

4 Robot Usage Process

If you are using the programmable robot for the first time, you need to configure the wireless access point first. Log in to your WeChat account, click on the mini program, enter "Search" and search for "Programmable Robot Wireless Configuration".

When using the WeChat mini program to configure the wireless for the robot, make sure that Bluetooth is turned on on your phone. "Programmable Robot Wireless Configuration" will list the programmable robots that have been found through Bluetooth search, and you can configure the wireless for any one of them.

After the programmable robot is powered on, the system starts up and obtains a dynamic IP address through the wireless module. Then, the IP address is broadcasted through the voice output module or can be queried through the WeChat mini program.

If you know the IP address of the programmable robot, you can access the online Scratch programming environment on the programmable robot through a browser on a computer or tablet. The address is https://IP_ADDRESS/scratch/, where IP_ADDRESS is the IP address of the Raspberry Pi. It should be noted that if you need to use the AI plugin, you must use the https protocol.

5 Artificial Intelligence Plugin

On the main interface of the Scratch learning environment, there is a button in the lower left corner that serves as the entrance for Scratch extension plugins. The Scratch AI plugin we are introducing in this article can also be accessed from here.

Currently, the implemented plugins include digit recognition, Pictionary, text recognition, face detection, object detection, image classification, speech training, and voice chat, covering the applications of AI in image, speech, and text.

In artificial intelligence applications, it is usually necessary to identify the detected objects. The pen provided by the standard Scratch plugin cannot meet this requirement, so the drawing tool here is needed. The drawing tool can draw lines, rectangles, circles, and images, which can be used with face detection, posture detection, and other plugins.

Computer vision is one of the main directions of artificial intelligence applications. Computer vision mainly analyzes and processes computer images and videos. The main purpose of the Shape and Background plugins is to provide image inputs for other artificial intelligence plugins. These images can be shape images or background images, which may come from local uploads, hand-drawn, or captured by a computer or robot camera.

Another typical artificial intelligence plugin is "Image Classification". In this plugin, the most important blocks are the training image block and the prediction image block,as shown in Fig.1..

.

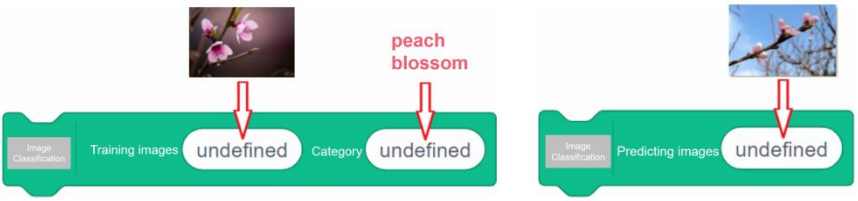

Fig. 1. Training and prediction image blocks of image classification plugin

The training image block specifies the category of the training image. The first parameter is the image to be trained, and the second parameter is the category to which the image belongs. For example, if the robot is learning about peach blossoms, the peach blossom image is used as the first parameter, and the second parameter is filled in as "peach blossom".

The prediction image block predicts the category of a given image. The parameter passed in is the image to be predicted. The prediction image block must be executed after the training is completed. The predicted result is obtained through the "predicted category" block.

6 Artificial Intelligence Applications

By using the Scratch AI plugins, different artificial intelligence applications in computer vision, computer speech, and text prediction can be achieved. Here, we will create a flower recognition robot that can recognize two different types of flowers: peach blossom and tulip,,as shown in Fig.2..

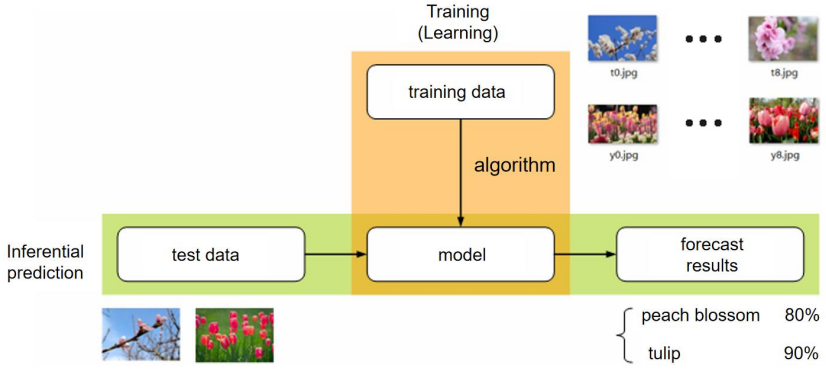

Fig. 2. the Training and Reasoning Process of Machine Learning

The production of the flower recognition robot uses machine learning methods. Specifically, we put the peach blossom and tulip images into their respective datasets, and put the image to be predicted into the test set, and then perform the training and inference processes.

The training process, also known as the learning process, is the yellow box part in the figure. We feed the data in the dataset to the robot for learning, telling the robot

that this image is a peach blossom, and that image is a tulip, and the robot "remembers" it and stores its "memory" in the model. This model is called a trained model.

After that, the robot can make predictions, which is also known as the inference process, which is the green box part in the figure. We feed the data in the test set to the robot, and the robot gives a prediction result based on the trained model. If the robot's prediction result is roughly consistent with the actual category of the image, then the robot's prediction is considered correct.

Next, we can use the above method to implement the flower recognition robot. For this purpose, four roles are designed, and message passing is implemented through broadcasting between different roles:

- Dataset1 (Dataset2) role: contains the peach blossom (tulip) dataset, which stores known peach blossom (tulip) image shapes. Upon receiving the "train dataset1" ("train dataset2") command, it sequentially trains the peach blossom (tulip) image shapes in it and reports the training process.
- Test set role: contains the image shapes to be predicted. Upon receiving the prediction command, it predicts the image shapes in the test set and reports the prediction result.
- Robot: it is the scheduling role, which sends training commands to Dataset1 and Dataset2 in turn, and finally sends a prediction command to the test set.

Upon receiving the broadcast message "train dataset1", the robot starts training sequentially. First, it switches to the first t0 shape, displays it, and then repeats the process 9 times because there are 9 shapes in total. Each time, it reads the name of the shape to be trained, and then performs the training. The training uses the "train image" block in the Image Classification plugin, with the shape image passed in and the category specified as peach blossom. After all the shape images are trained, they are hidden. Finally, the robot reads out "training completed" and reports the number of samples included.

7 Conclusion

This article introduces programmable robots and the built-in Scratch AI plugins. Through these plugins, it helps to understand the general process of artificial intelligence, such as data collection, model training, and prediction inference. There are multiple possibilities for prediction results, which are reflected in accuracy, and this should be noted when applied in specific applications.

Based on these plugins, students can also use their imagination to create their own AI applications with independent creativity. To further understand the principles behind them, one can learn about web crawling, data preprocessing, data visualization in Python programming, as well as linear regression, logistic regression, K-means, neural networks, and other aspects in machine learning and deep learning.

Acknowledgments

This research achievement is supported by named the "Shanghai Publishing and printing college high level talent start-up fund project (2022rcky04)", which is the achievement of the project.

References

1. EO Intelligence. 2019 Global Artificial Intelligence Education Industry Research Report, 2019.
2. Li Xiaoxiao. 36Kr - Research report on the foundation education industry of artificial intelligence, 2019.4.
3. J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond. The scratch programming language and environment. Trans. Comput. Educ., 10:16:1–16:15, November 2010.
4. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: A system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16) (GA, 2016), USENIX Association, pp. 265–283.
5. Matt Richardson and Shawn Wallance, Getting Started with Raspberry Pi (USA, O'Reilly Media, 2012).
6. Li Dandan, Jiang Yu. Introduction to artificial intelligence instance development based on NVIDIA Jetson platform. Harbin: Harbin Institute of Technology Press, 2019.
7. Akbar Karimi, Leonardo Rossi, and Andrea Prati. AEDA: An Easier Data Augmentation Technique for Text Classification[C]//In Findings of the Association for Computational Linguistics: EMNLP 2021:pages 2748–2754.
8. Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning representations by maximizing mutual information across views[J]//In Advances in Neural Information Processing Systems, pp. 15509–15519, 2019.
9. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding [C] //In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers),2019: pages 4171–4186.
10. Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks[J]//In Advances in neural information processing systems,2012: pp. 1097–1105.