Puck tracking system for aerohockey game with YOLO2

A E Tolmacheva, D A Ogurcov and M G Dorrer

1Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, prospect “Krasnoyarsky rabochy “31, Russia

2E-mail: mdorrer@mail.ru, dim_ogu@mail.ru

Abstract. This article is devoted to the preparation of the YOLO2 convolutional neural network for use in the robot artificial intelligence tracking module in air hockey competitions in the human-robot or robot-robot category. Such developments are unique in the sphere of artificial intelligence and robotic technologies. The task was to collect and prepare educational material, as well as further training and testing of the convolutional neural network to create a “vision” and further predict the trajectory of a fixed object. For this, the YOLO2 model was used, created by the developer Ali Farhadi in the low-level language C [1]. As a result of testing in the next game, the results of the correct detection of the object in the range of 80% were obtained, despite the lack of a standard camera position and poor image quality. In the future, the created system can be used in the further development of the AI robot systems and for creating tactics of behavior in various air hockey matches.

1. Introduction

Today, international competitions are taking place among enthusiastic programmers around the world, creating unique algorithms that allow robots to fight on a par with humans. One striking example is air hockey. Two robots or a robot-human compete with each other for 2 halves. Victory counts in speed, tactics accuracy and most importantly in score points. The unique technologies used to calculate and predict the trajectories of the object (puck) are also taken into account. These events, confronting a robot and a human, can improve the design methods of automated systems, the creation of new algorithms in the field of robotics.

In this project, the problem of using robots and artificial intelligence (in particular, artificial neural networks) in various games in the categories of human-robot, robot-robot was solved.

Computer intelligence Pluribus [2] from the developers of Carnegie Melon University played poker and for the first time in history beat five professional people. For 12 days, Pluribus took part in 10 thousand parties, for the victory in each of which received 5 dollars. By the end of the experiment, AI began to earn up to 1 thousand dollars per hour and play better than any opponent. The developers explained that the computer’s mathematical strategies were more effective than human bluffing, and less power was required for computing than its technological predecessors. In the future, they intend to use this system in the field of cybersecurity, protecting people from scammers.

Japanese researchers [3] have developed a robot that evaluates and predicts the actions of an opponent. The robot’s camera captures 500 frames per second - this way the device monitors the opponent’s hand and puck movements, determines whether the player is attacking or defending, and then, based on these data, draws up his own strategy. During the game, the robot adapts to the style of
the opponent. This robot was part of a large project to create an interactive android that analyzes human movements.

The Pick Your Prototype Challenge [4] was attended by a project of the Spanish engineer Jose Julio, who assembled cheap robotic air hockey based on the Arduino Mega 2560 platform from improvised means. The robot is able to calculate the trajectory of its movement and hit the projectile. The robot can play in three modes: counterattack and defense, attack, defense. The level of "intelligence" of the machine can be changed under the skills of the human opponent. In order for the system to recognize and track the puck, installed a camera from a Playstation 3 connected to a PC. The project is completely open and the program code with full instructions lies in the GitHub system.

The current stage of the development of the theory and practice of ANN in terms of working with images is characterized by the fact that they began to recognize images better than a person does with the naked eye. This result was achieved through the use of convolutional neural networks and deep learning algorithms. Such networks are often used in the field of robotics and machine automation.

From the above examples, two important conclusions can be distinguished:

- Neural networks are necessary in the field of robotics, as this allows the development of artificial intelligence algorithms, devoting part of interactivity with humans.
- Similar competitive events created on the basis of games between a human and a robot allow finding new methods for solving automation of systems.

At the same time, in known solutions, the use of convolutional neural networks based on the Darknet framework was not encountered. Most often, researchers used computer vision algorithms. In this work, implemented solution that differs from the well-known ones in that it uses YOLO2 technology, which allows you to capture a fast-moving object in the frame even when the low stream quality, which allows you to save on equipment and in the absence of fixed points of view on the playing area of air hockey.

2. Methods

The problem of real-time image recognition is currently represented by two approaches: the first, in which regions of the frame are proposed and classified in turns, is called Region Proposes and the second, in which all objects are immediately detected on the image, is called Single Shot. The first approach to image processing is used in such neural networks as: R-CNN, Fast R-CNN, Faster R-CNN [5] And the second in: YOLO CNN [6], SSD [7]. Artificial neural networks using recognition by region works rather slow when determining objects in images. For lighter and more mobile platforms, systems such as Single Shot CNN are most suitable, as they are faster.

When performing the objectives of this study was used the architecture of the convolutional neural network Darknet model YOLO2. DarkNet stores trained coefficients in a format that can be recognized using various methods on different platforms. In this approach, the neural network predicts bounding boxes and class probabilities, applying to the full image. YOLO2 is one of the fastest algorithms and is suitable for working in real time, but has low accuracy on too large and too small objects (average accuracy (mAP) —63.4%).

In the YOLOv2 modification, a number of improvements have been added that give a significant increase in the speed and accuracy of the algorithm, while the accuracy has reached the accuracy of Faster R-CNN. The most significant improvements are:

- Batch Normalization gives an accuracy increase of 2%. When training images are fed into the neural network in batches, respectively, the values of the weights are updated after processing the bundle. Batch normalization leads to a significant improvement in convergence, eliminating the need for other forms of regularization. By adding batch normalization at all convolutional levels in YOLO, we got more than 2% improvement in mAP. Normalizing the batch also helps streamline the AI model;
- High Resolution Classifier. Fine-tune the resulting network detection. Used a classifier trained on the ImageNet dataset. This high-resolution classification network provides an increase of almost 4% mAP;
- Dimension Clusters + Direct location prediction. The use of dimensional clusters along with direct prediction of the center location of the framing window improves YOLO by almost 5%.

For the process of training a neural network, a system with current technical characteristics was used:

**Table1.** Server Specifications.

| Component Type | Model                                      |
|----------------|--------------------------------------------|
| CPU            | Intel i7-8700k                              |
| GPU            | GeForce GTX 1070 Ti                        |
| Memory RAM     | Corsair Dominator Platinum 32GB (2×16)     |
|                | 3000Mhz                                    |

Two training games with a total duration of five minutes were held to collect material for the future Dataset. The degree of illumination and the random position of the camera were taken into account, as well as small artifacts in the form of interference associated with the cheapness of the USB camera used.

For training, set of images must be created for which there is an appropriate set of markup. In this implementation, for an image with a name, for example images01.jpg, there must be a text file with the same name, but with the extension .txt, i.e. images01.txt.

This file should contain a set of lines of the form: "class" "x" "y" "width" "height".

- "class" - natural numbers from zero to (classes-1).
- "x" "y" "width" "height" - fractional numbers corresponding to the width and height of the image from 0.0 to 1.0 - i.e. correspond to the scale of the entire image.
- "x" = "x coordinate" / "image width"
- "height" = "object height" / "image height"
- "x" "y" is the center of the square

To collect high-quality diverse material, it was decided to use all the training material obtained as a result of the game. This was done by two people to reduce the processing time of the frame. The obtained and processed photographs according to YOLO standards were placed on training and trained for 1.5 hours with a number of epochs approximately equal to 5000.

The resulting neural network model was transferred to the Raspberry pi3 platform, which is responsible for analyzing the video stream and forming the response of the robot holding the striker.

The problem of video processing was posed as one of the key in this task.

The module responsible for processing the video stream and the response in the form of developing a robot game strategy is used on the Raspberry Pi3 microcomputer platform.

For microcomputer families such as raspberry pi, a yolo-tiny detection system has been created. This version allows you to increase the processing speed of the frame and obtain the detection area of the area where the puck is located.

**Table2.** Technical specifications Raspberry pi 3.

| Component Type | Model                                      |
|----------------|--------------------------------------------|
| CPU            | 64-bit 4-core ARM Cortex-A53 with a clock frequency of 1.2 GHz on a single-chip Broadcom BCM2837 chip; |
| RAM            | 1ГБ LPDDR2 SDRAM                              |
3. Results
Figure 1 shows the settings of the YOLO2 neural network for one class "puck".

Figure 2 shows the learning process and a graph of the loss error. Learning has been stopped at 0.066 AVG loss.

Figure 3 shows examples of a neural network operation on a video clip shot on a another match. The camera can be located at any viewing angle.

Figure 4 shows the results of the correct answer at high FPS.

Figure 1. Darknet settings for single class «puck».

Figure 2. Stages of neural network training. Avg loss chart.
Figure 3. Example of operation of the YOLO2 neural network with different camera positions.

4. Discussion
In the images, you can see that the trained model captures and holds the areas of the puck in motion, ignoring the presence of bits of the same color and similar shape. Figure 8, 9.

Figure 4. The average percentage of detection of objects by the YOLO2 neural network.
5. Conclusion

Thus, as a result of the work performed, a system for automatic recognition of objects with a coordinate tracking system has been developed. This model allowed us to stably process the video stream with a frame rate of 30 FPS. The percentage of accuracy of this trained network is 80% on specially prepared video material.

The prospect of this work is the development of a module for tracking enemy tactics using deep neural networks and the creation of winning strategies for a game robot for participation in international competitions.

References

[1] YOLO: Real-Time Object Detection (Electronic Materials) Access mode: https://pjreddie.com/darknet/yolo/
[2] Brown N 2019 AI for multiplayer poker Science 365 6456
[3] Matsushita N S, Ozeki T and Nonami K 2013 Hierarchical Processing Architecture for an Air-Hockey Robot System IEEE International Conference on Robotics and Automation (ICRA)
[4] Julio J 2019 Air Hockey Robot (a 3D printer hack) (Electronic Materials) Access mode: https://github.com/JJulio/AHRobot
[5] Ren S, He R and Girshick J 2017 Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks IEEE Transactions on Pattern Analysis and Machine Intelligence 39 6
[6] Redmon J and Divvala S R 2016 You Only Look Once: Unified, Real-Time Object Detection IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
[7] Liu W and Anguelov D 2016 SSD: Single Shot MultiBox Detector European Conference on Computer Vision 21-37