Auxiliary Use of YOLOv5 in FPS Shooting Competitive Games

Hao Tian, Jiadun Qi, Yong Sun, Zhepei Zhang, Ruohan Wen, Cheng Huang and Kao-kin Zao
Department of Information Engineering
The Chinese University of Hong Kong
Hong Kong, China
johnkzao@cuhk.edu.hk

***Abstract*—This paper highlights that we use a deep learning technique to assist our players' win rate in competitive FPS games. Our deep learning approach is based on YOLOv5, a derivative of YOLO, an object detection model and we have made certain improvements based on YOLOv5. The games used as experimental objects are CSGO, CF, and PUBG. They are both first-person shooter games. The experimental results show that our method improves the accuracy of marksmanship when assisting players in competitive shooting. At the same time, we also marked the enemies behind the obstacles. Whether this method is based on real players, it has a certain degree of assistance. In the future, we will combine reinforcement learning to generate an AI player to cooperate with our game assistant for experiments. At the same time, we will continue to improve our model based on the existing auxiliary shooting technology.**

***Index Terms*—deep learning, YOLOv5, game assisted shooting**

I. Introduction

With the development of games, e-sports has gradually entered people's field of vision. Many games, such as League of Legends, Overwatch, etc., have attracted people's attention and participation. Among them, first-person shooter games in the fps category are extremely popular among the crowd. There are many games represented here: Overwatch, Crossfire, CSGO, PUBG, etc. However, these games themselves have certain technical thresholds, and the experience of novice players in these games is not very good. They are easily killed by skilled players in the game because of their skill. And it was born with the game plug-in, that is, third-party illegal software. With the help of these software, some players perform far beyond ordinary people in the game. At the same time, with the upgrading of computer science technology and corresponding hardware, more and more new game auxiliary software has also entered the market, among which artificial intelligence is a very hot topic.

Artificial intelligence technology is currently a very hot computing technology. It forms an intersection with many projects in real life [1][2][3][4]. Likewise, this technology can also be used in eSports and everyday gaming, especially in highly competitive FPS games. Many first-person shooter games have assisted aiming if this technology is used, which can help novice players to improve the game experience well. This involves the method we will use in this assisted aiming technology. Deep learning [5] is a branch of artificial intelligence. Deep learning is to learn the inherent laws and representation levels of sample data, and the information obtained during these learning processes is of great help to the interpretation of data such as text, images, and sounds. Its ultimate goal is to enable machines to have the ability to analyze and learn like humans, and to recognize data such as words, images, and sounds. It has achieved many results in search technology, data mining, machine learning, machine translation, natural language processing, multimedia learning, speech, recommendation and personalization technology, and other related fields. Deep learning enables machines to imitate human activities such as audio-visual and thinking, solves many complex pattern recognition problems, and makes great progress in artificial intelligence-related technologies.

The deep learning methods of auxiliary shooting used in FPS games are mostly related to computer vision technology. In computer vision technology, there are very classic target detection methods, such as R-CNN [6]. R-CNN will mark a frame at the target object to indicate the position of the target in the picture or the surrounding environment. If we can apply this technology in the game, for example, mark the head position of the character in advance, so as to A good head position reference is provided for novice players, which can help novice players adapt and experience the game in the early stages. But R-CNN is an early computer vision model, and the file it trains is too large. If you encapsulate it and then assist the game, it will seriously affect the system performance and game experience, and even cause the system to crash. And its derivative models, such as Fast R-CNN [7], Faster R-CNN

Hao Tian is the first author. He is currently pursuing a master's degree in Information Engineering at the Chinese University of Hong Kong. He will go to the Hong Kong Polytechnic University in 2022 to study for a doctorate in engineering, with research directions in fashion design and computer vision technology. His E-mail is 1155169008@link.cuhk.edu.hk.

Jiadun Qi, Yong Sun, Zhepei Zhang, Ruohan Wen and Cheng Huang are currently pursuing a master's degree at the Department of Information Engineering, Chinese University of Hong Kong. Their E-mails are respectively {1155161048, 1155169190, 1155160918, 1155166052, 1155152552} @link.cuhk.edu.hk.

Professor John (Kao-kin Zao) is currently a Professor of Practice in Information Engineering at the School of Engineering, Chinese University of Hong Kong, and the corresponding author of this paper. His research interests are Internet Engineering (IoT & Edge Computing); Wireless Communications and Networking.

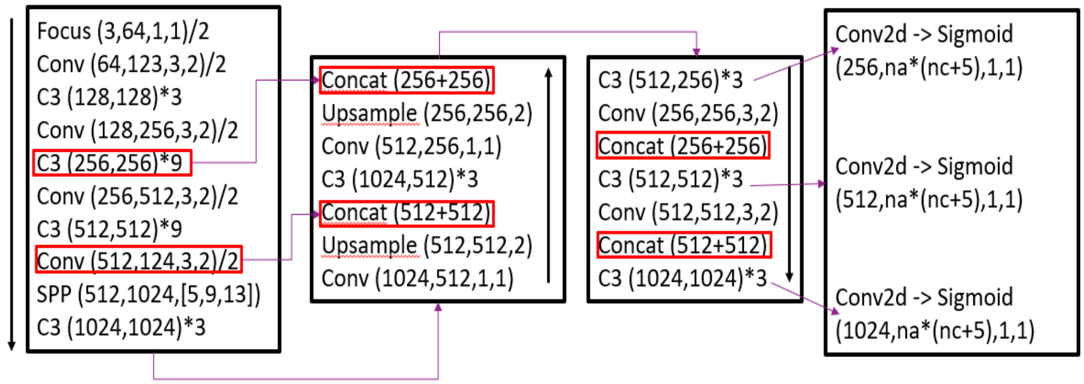

Fig. 1. The structure of YOLOv5.

[8], and Mask R-CNN [9], also face this problem.

But during our experiments, we found that the author of R-CNN has another model that can deal with this problem very well. This model is YOLO (You Only Look Once) [10]. Not only that, the YOLO series has many upgraded versions so far, such as YOLOv2 [11], YOLOv3 [12], YOLOv4 [13], and YOLOv5 [14]. Compared with the previous YOLO series, YOLOv5 has a huge difference, especially the size. YOLOv5 is only a few tens of MB, much smaller than the hundreds of MB of YOLOv4. Not only that, it surpasses the previous version in speed while maintaining accuracy.

Based on the lightweight framework of YOLOv5, we use it as the native model of FPS games, and based on this, we develop an FPS auxiliary software that can be used in popular shooting games, such as CF, PUBG, and CSGO. We also added the previous model to the experiment to conduct a certain comparison and analysis. Because of this, based on this effect, we will continue to improve our method in the follow-up, so as to make a shooting game auxiliary software that can help novices well.

II. Related Work

A. YOLOv5

YOLOv5 has made further improvements on the basis of the YOLOv4 algorithm, and the detection performance has been further improved. The model structure is shown in Fig. 1. But in essence, YOLOv4 and YOLOv5 are basically similar in structure, but slightly different in details.

YOLOv5 increases the focus structure The core of the Focus structure is to slice the picture, so that the features can be collected more accurately and abundantly. Furthermore, in YOLOv4 only the CSP structure is used in Backbone, while in YOLOv5 two different CSPs are used in Backbone and Neck. In Backbone, CSP1_X with residual structure is used. Because the Backbone network is deep, the addition of residual structure makes the gradient value enhanced when backpropagating between layers, effectively preventing the gradient caused by the deepening of the network. disappear, and the resulting feature granularity is finer. Using CSP2_X in Neck, compared with pure CBL, the output of the backbone network is divided into two branches, and then concat is used to strengthen the network's ability to integrate features and retain richer feature information.

B. Game Assisted Shooting

There are many examples of deep reinforcement learning combined with games [15][16]. In these papers, deep reinforcement learning is used directly as a method to generate an AI player. This AI player can perform player-like actions such as shooting, moving, crouching, jumping, etc. In our paper, they are not the protagonists this time, but we will use the ”AI people” they generate. The paper [17] uses a combination of two deep learning methods to assist the player to accurately locate the enemy's position, and will provide feedback and interaction according to the player's operating habits. The paper [18] uses a deep learning method to make the characters in the game move flexibly. This may help players avoid damage in FPS shooting games, but it will also bring some difficulty to aiming. The paper [19] presents the first two editions of Visual Doom AI Competition. To play well, the bots needed to understand their surroundings, navigate, explore, and handle the opponents at the same time. These aspects, together with the competitive multiagent aspect of the game, make the competition a unique platform for evaluating the state-of -the-art reinforcement learning algorithms.

At the same time, we also refer to many improved methods based on YOLOv5 itself [20][21][22]. Most of them are based on the structure or method of YOLOv5 with certain improvements. And we can adjust and package YOLOv5 based on these improvements to meet the needs of the game.

III. Methodology

In this paper, we have used YOLOv5s, which is the smallest one among YOLOv5 series. YOLOV5s is a one-stage object detection deep learning method that outputs the position and category confidence of the object box at one time. Yolov5s is augmented by Mosaic data. Four images are read at a time. Flip, scale, and change the color gamut of the four images separately, combining images and frames. Since we only need

Fig. 2. The experimental results of our method.

to retrieve the character model from the terrain in this project, and do not need to correct and adjust other items or details, we have made certain changes to the structure of the model. We have replace Bottleneck structures in the original one with Convolutional layer with halved linear transformation. This reduces the parameters by half, such as the convolution kernel. In this way, the convolution calculation uses a smaller number of convolution operations than the original, using depth separation convolution, and then transforms redundant features from the feature map generated above, and then the feature map obtained in the two steps above is concat and output, and sent to the subsequent link.

At the same time, due to the single type of training target and the high similarity of most of the images, in order to prevent over-fitting, we also adjusted its parameters to a certain extent as shown in below Table I. Reducing the epoch is to prevent overfitting; changing the initial (lr0) and final (lrf) learning rates is so that the network can converge to the global minimum without gradient problems; IoU is adjusted to make the model better The threshold candidate box screening.

TABLE I
PARAMETERS OF OUR METHOD

parameters	origianl	ours
epoch	500	100
lr0	0.01	0.005
lrf	0.02	0.01
IoU	0.5	0.7

[a]Nouns are abbreviated.

IV. Dataset Pre-processing and Equipment

A. Dataset and Pre-processing

Regarding the original dataset, we collected high-definition pictures from the game, and the resolution is based on the resolution of the game itself. For CF, PUBG and CSGO, we collected 500 images as the training set, and 500 images as the test images.

These images are marked by the software Labelme, and then the generated json file is converted into a txt pixel collection file, and finally stored in the corresponding training data set.

B. Equipment

When training and testing, we do it on the server side. Its hardware is 500GB of solid-state storage, 16GB of RAM, and a 2080Ti graphics card. For the video streaming test, we load its server network ip address and its port through the local VLC video streaming software to view the effect.

As for the tests of the games themselves, we all completed them on students' PCs, because most of these games are not supported on linux. The student's test computer is 128gb+1tgb solid state, 16gb memory, and 3060 graphics card.

V. Experimental Results

As shown in Fig. 2, we gets results based our method from the real game situation. Our method can detect the enemy and mark it with a candidate box at the position of the enemy in our view. At the same time, it can also be seen from the picture that even the character model behind the obstacle can be detected with a red frame. This approach can help novice players predict where the enemy will appear in advance.

As shown in Fig. 3. there is no red box in the game screen itself, but the picture we see on the visual side is a red box, that's because when we intercept the data, it's not when we send the data packets to detect, but to wait for the game data After its server processing, the transmission back to the local is for detection. So it will appear that when we record video with the game's own recorder, there is no red box, but we can see it when we use external video recording software.

As shown in Table II, we get different evalaution which focus on model itself, among different games , and we're also adding a test for Overwatch. Among them, we tested prototype models of game characters without considering their skins. The test data is based on recorded video rather than live tests. *other* in the game represents characters that are not human or human-like, such as zombie dogs, or monsters. The calculation method of Accuracy is the ratio of the number of correctly

TABLE II
ACCURACY RECOGNITION COMPARISON OF DIFFERENT GAMES

Method	category	CF	CSGO	PUBG	Overwatch
YOLOv5	human	93.86%	94.89%	94.99%	91.87%
	Zombie	92.12%	-	95.01%	-
	aniaml	-	-	-	98.86%
	robot	-	-	-	95.61%
	other	95.89%	-	-	95.87%
SSD[23]	human	92.45%	94.96%	95.01%	92.31%
	Zombie	91.89%	-	95.86	-
	aniaml	-	-	-	98.75%
	robot	-	-	-	96.12%
	other	94.19%	-	-	94.36%
Faster R-CNN	human	91.86%	96.02%	93.11%	90.86%
	Zombie	92.45%	-	93.09%	-
	animal	-	-	-	91.56%
	robot	-	-	-	93.67%
	other	92.13%	-	-	95.14%
Our method	human	94.85%	95.63%	96.34%	95.42%
	Zombie	94.56%	-	96.21%	-
	anmial	-	-	-	98.91%
	robot	-	-	-	96.38%
	other	96.55%	-	-	97.83%

[a] - Indicates that the game does not have an object model of this type.

Fig. 3. The real environment of the game and the picture from our perspective.

judged categories to the total number of categories. Its formula form is as follows:

$$Accuracy = \frac{TP}{TP + FN}$$

In general, our improved method has a good improvement in the accuracy index compared with the previous method, and when identifying some reference objects, such as road signs and robots. Our method is less prone to false positives, and our detection of tiny objects (distant objects) is more accurate than previous models.

CONCLUSION

In this paper, we use the YOLOv5 model and make some improvements to make it a shooting aid for FPS games. It assists the player's shooting effect and improves the game experience for novice players. Even targets behind obstacles, we can detect them to help novice players predict their positions and actions in advance, which can help novice players become familiar with shooting. In the future, we will use reinforcement learning to generate AI players for testing, and we will also conduct functional expansion research on our model, such as research on head detection and lock tracking. At the same time, we have also seen some problems: when the color of the target itself is not much different from its background, for example, the target and Beijing are both black, and they overlap in some places, and sometimes they cannot be detected. We will explore and improve this in future research.

ACKNOWLEDGE

Thank you very much for your wonderful cooperation in this class of IEMS 5709. We not only completed intra-group cooperation, but also achieved good cross-group cooperation between groups. We also congratulate us on the success of this project!

REFERENCES

[1] Liu Xian, "Artificial intelligence and modern sports education technology," 2010 International Conference on Artificial Intelligence and Education (ICAIE), 2010, pp. 772-776.

[2] A. S. Ahmad, "Brain inspired cognitive artificial intelligence for knowledge extraction and intelligent instrumentation system," 2017 International Symposium on Electronics and Smart Devices (ISESD), 2017, pp. 352-356.

[3] N. Wang, Y. Liu, Z. Liu and X. Huang, "Application of Artificial Intelligence and Big Data in Modern Financial Management," 2020 International Conference on Artificial Intelligence and Education (ICAIE), 2020, pp. 85-87.

[4] M. Hachem and B. K. Sharma, "Artificial Intelligence in Prediction of PostMortem Interval (PMI) Through Blood Biomarkers in Forensic Examination–A Concept," 2019 Amity International Conference on Artificial Intelligence (AICAI), 2019, pp. 255-258.

[5] D. Goularas and S. Kamis, "Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data," 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), 2019, pp. 12-17.

[6] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580-587.

[7] R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448.

[8] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017.

[9] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988.

[10] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.

[11] J. Redmon and A Farhadi, "YOL09000: better faster stronger[C]", Proceedings of the IEEE conference on computer vision and pattern recognition., pp. 7263-7271, 2017.

[12] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement", Computer Science, 2018.

[13] Alexey Bochkovskiy, Chien-Yao Wang and Hong-Yuan Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection", 2020.

[14] T. F. Dima and M. E. Ahmed, "Using YOLOv5 Algorithm to Detect and Recognize American Sign Language," 2021 International Conference on Information Technology (ICIT), 2021, pp. 603-607.

[15] J. Bergdahl, C. Gordillo, K. Tollmar and L. Gisslén, "Augmenting Automated Game Testing with Deep Reinforcement Learning," 2020 IEEE Conference on Games (CoG), 2020, pp. 600-603.

[16] K. Shao, D. Zhao, N. Li and Y. Zhu, "Learning Battles in ViZDoom via Deep Reinforcement Learning," 2018 IEEE Conference on Computational Intelligence and Games (CIG), 2018, pp. 1-4.

[17] A. P. Poulsen, M. Thorhauge, M. H. Funch and S. Risi, "DLNE: A hybridization of deep learning and neuroevolution for visual control," 2017 IEEE Conference on Computational Intelligence and Games (CIG), 2017, pp. 256-263.

[18] S. Nilwong and G. Capi, "Outdoor Robot Navigation System using Game-Based DQN and Augmented Reality," 2020 17th International Conference on Ubiquitous Robots (UR), 2020, pp. 74-80.

[19] M. Wydmuch, M. Kempka and W. Jaśkowski, "ViZDoom Competitions: Playing Doom From Pixels," in IEEE Transactions on Games, vol. 11, no. 3, pp. 248-259, Sept. 2019.

[20] J. Ieamsaard, S. N. Charoensook and S. Yammen, "Deep Learning-based Face Mask Detection Using YoloV5," 2021 9th International Electrical Engineering Congress (iEECON), 2021, pp. 428-431.

[21] E. Cengil, A. Çinar and M. Yildirim, "A Case Study: Cat-Dog Face Detector Based on YOLOv5," 2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), 2021, pp. 149-153.

[22] L. Zheng, X. Wang, Q. Wang, S. Wang and X. Liu, "A Fabric Defect Detection Method Based on Improved YOLOv5," 2021 7th International Conference on Computer and Communications (ICCC), 2021, pp. 620-624.

[23] W. Liu, D. Anguelov, D. Erhan, S. Christian, S. Reed, C.-Y. Fu, et al., "SSD: single shot multibox detector", ECCV, 2016.