Using artificial intelligence for object recognition in three-dimensional video games

David Livshits, Georgi Hristov

The report explores the use of artificial intelligence for object recognition in three-dimensional video games. A methodology for collecting a sufficient volume of images, which are subsequently analyzed and processed by adding key points, is presented. The set key points serve to train the artificial intelligence that is to be used. Two libraries for training and working with artificial intelligence are examined and their applicability is analyzed. Proposals are made for integrating the achievements of artificial intelligence in object recognition in video games into various socio-economic spheres.

Keywords – artificial intelligence, object recognition, three-dimensional video games, object recognition.

Използване на изкуствен интелект за разпознаване на обекти в триизмерни видеоигри (Давид Лившиц, Георги Христов). *В доклада се разглежда тематика свързана с използването на изкуствен интелект за разпознаване на обекти в триизмерни видеоигри. Представена е методика за събиране на достатъчен обем от изображения, които впоследствие се анализират и обработват, чрез добавяне на ключови точки. Поставените ключови точки служат за обучението на изкуствения интелект. Разгледани са две библиотеки за обучението и работата на изкуствения интелект, като е направен анализ на приложимостта им. Направени са предложения за интегрирането на достиженията на изкуствения интелект за разпознаване на обекти във видеоигри в различни социално-икономически сфери..*

Introduction

Nowadays, artificial intelligence technologies continue to evolve and surprise us with new possibilities. These systems constantly show us new ways of object and creature recognition in real-time. This can have a significant impact on our way of life and it can offer us more efficient and innovative solutions in various fields. One of the recent innovations is in the field of artificial intelligence systems that can target objects or entities from a display in real-time, utilizing a large number of screenshots that pass through neural networks. This technology has numerous applications in various areas of life, such as security, healthcare, the automotive industry, and others. In this report, we will explore these systems, their capabilities, advantages, and some limitations.

Methods

The selection of algorithms in artificial intelligence is a pivotal factor in the development of advanced and efficient gaming artificial intelligence. In most games, there is a constant battle between human players and artificial intelligence opponents. To create a more enjoyable experience for human players, it is imperative to explore different methods for choosing the optimal algorithms that can enhance the efficiency and success rate of computer-controlled players. [1]

Most commonly, in artificial intelligence, a neuro-evolution algorithm is used [2]. However, the problem is that this won't work for all purposes. For example, if we want such a machine intelligence to play a game, then first it must be taught as it is done with a young child. Typically, a copy of the game is created, in which sensors and the ability to create 10-20 copies of players are added. Then generations are created, where the more successful genomes are rewarded, while the less successful ones have their rewards reduced. And thus, in this way, in the end, one genome remains capable of performing its task very well.

But what should we do if our game is three-dimensional, and the goal is to spot the enemy and aim at them, like in CS:GO, for example? Of course, we can try to create a copy of it, but it would be desirable for the artificial intelligence to play the actual game and show good results.

If we think about it, we don't need many things for a successful game. For that purpose, our program needs to be able to do precisely three things:

1. See the game.
2. Be able to aim and shoot.
3. Be able to navigate on the map (which is actually not necessary since we can control the player ourselves).

Wu, Yi, Ma and Chen [1] highlight the main techniques for integrating artificial intelligence into 3D games, noting that they can be used in the creation and visualization of various game elements. These techniques are:

- Ad hoc behavior authoring - this approach is beneficial for developing dynamic and responsive AI capable of adapting to changing game conditions [3].
- Tree search algorithms - In video game AI, various tree search algorithms are utilized to enhance both gameplay logic and visualization.
- Evolutionary computation – it can be utilized to enhance the performance and capabilities of AI characters within a game.
- Machine learning - Incorporating machine learning into game AI allows for the adaptation and improvement of AI overtime while also enabling it to learn and respond to the specific needs and preferences of the player. Machine learning techniques including supervised learning, unsupervised learning, reinforcement learning, and deep learning play a crucial role in achieving these goals.
- Neural networks - By incorporating neural networks into game AI, AI characters can continuously learn and adapt over time.

The conclusion of the authors [1] is that they have extensively examined various artificial intelligence applications in the visualization of video games. These applications encompass the use of machine learning algorithms for tasks such as character motion, terrain generation, and lighting effects. They have also delved into the benefits, challenges, and potential applications of artificial intelligence in this field. One notable trend is the utilization of artificial intelligence to enhance the visual realism of video games, achieved through machine learning algorithms for generating more lifelike graphics and optimizing the rendering process. Another trend involves the creation of more immersive and interactive gaming environments through artificial intelligence technologies. Additionally, there is a growing inclination to harness artificial intelligence for crafting more realistic and intelligent in-game characters, enabling them to make decisions and adapt to dynamic gaming scenarios through machine learning algorithms. The future outlook for artificial intelligence in video games points toward increasingly sophisticated and lifelike artificial intelligence models and expanded applications of machine learning.

Artificial Intelligence Vision

To enable the game to see, a large number of screenshots can be taken. For this purpose, Python code is written, incorporating the OpenCV library [4] for computer vision and the mss library [5] for capturing screen screenshots. Upon running the code, the game captures 20 frames per second, which is already quite good, but further improvement can be achieved by introducing multithreading [6], where the processes of capturing and saving screenshots are separated in the code. With this approach, the program can see 30 frames per second.

Aiming and shooting

The next step is to teach the artificial intelligence how to aim and shoot. While it may sound easy in theory, in practice, it is necessary to teach it how to differentiate between player objects and, ideally, be able to distinguish between opponents and teammates. There are two options to solve this task:

1. Using OpenCV and its HAAR Cascade [7].
2. To train artificial intelligence through conditional TensorFlow.

The more convenient way would be through TensorFlow. TensorFlow is a free and open-source software library for machine training and artificial intelligence. It can be used for a variety of tasks, but it has a specific focus on training of deep neural networks.

A neural network is a machine learning algorithm inspired by the structure and functions of the human brain. It comprises interconnected "neurons" that process and transmit information. In the realm of game artificial intelligence, neural networks can be utilized to craft intelligent and adaptable artificial intelligence characters. These characters learn from their experiences and adjust to evolving game conditions. By training neural networks to recognize patterns in data, such as images or sounds, artificial intelligence characters can respond to various stimuli within the game environment. For instance, they can distinguish between different types of terrain or specific objects, thus enhancing their capacity to engage with the game world realistically and immersively. [1]

In the given case, it will help us teach artificial intelligence to distinguish between opponents and teammates [8].

Object recognition

In order to teach artificial intelligence to recognize objects, first it is necessary to show these objects to it, preferably in a large quantity. For this purpose, numerous screenshots of CS:GO are taken. Around 700 screenshots are obtained, which will serve as the training data.

However, just images alone won't be enough. With the help of a special program, each image needs to be manually opened and special points need to be placed to mark the desired objects in those images. This process takes several hours, with the marked objects being categorized into 4 different groups: 2 groups for the whole body and 2 specifically for the head. Afterwards, all this data is converted from the XML format to CSV format, which is then used to create the so-called TFRecord [9] file. It is also essential to create a label map since artificial intelligence works with numbers, while the group labels are in alphabetical form.

Lastly, a configuration file for the artificial intelligence needs to be written, specifying the number of classes and providing the paths to all the previously generated files.

Now the process of training artificial intelligence can be initiated. The speed of training depends on the processor and/or graphics card used.

After the training, screenshots from the game are presented, and the AI accurately identifies the players. For convenience, bounding boxes with class names are also visualized. The artificial intelligence precisely indicates the locations of opponents and teammates, and if possible, draws bounding boxes around their heads (Fig. 1). This information is sufficient for the artificial intelligence to aim effectively.

Practice

Since the artificial intelligence can already accurately identify players, it is necessary to launch CS:GO and test it in-game. However, an unexpected problem arises - when playing the game, the number of screenshots sharply drops to 10.

And, of course, with such a low number of frames nothing can be done, the frames are simply not enough and it cannot react in time to the enemy.

The problem most likely stems from the graphics card, specifically: The artificial intelligence process utilizes almost 100% of the graphics card resources, and if the game, which also utilizes the graphics card, is running simultaneously, it means that fewer resources are available for the artificial intelligence, resulting in a significant decrease in the number of processed frames.

Fig.1. Artificial intelligence for object recognition in three-dimensional video games.

This problem can be solved in two ways. The first approach is to add another graphics card to the computer, so that while one card handles the game rendering, the other one can be dedicated to processing the frames from the neural network. The second approach is to simply switch to a faster artificial intelligence library. Such a library exists, called YOLO (You Only Look Once) [10], which is considered the fastest artificial intelligence for real-time object detection.

However, there is another possible reason for the low number of processed frames per second, and that is the high resolution of the images.

Decreasing the resolution

The first step is to create Python code that analyzes all the output images from the neural network and provides detailed statistics. It turns out that more than 85% of all images have a resolution higher than 1366 by 768 pixels. This is problematic because artificial intelligence only needs something like 400 by 400 pixels or even smaller for its work. The solution is to reduce the resolution of the images to the required size. While it may seem simple, there is an important nuance. Do you remember that there is an XML file corresponding to each image that stores the data of the special points? These points are necessary for the artificial intelligence to understand the location of opponents during training and be able to differentiate them. The problem is that the data of these points is stored with respect to specific image dimensions, not in percentage ratios. For example, if an image has a resolution of 1 000 by 1 000 pixels, the positions of the points in its XML file will be specific to an image of size 1 000 by 1 000. Therefore, simply changing

the resolution of the images is not enough. It will be necessary to update the XML files by writing the new positions of the points, taking into account the new resolution. Although this process may not be too complex, it does make the integration code a bit larger. This storage format for points used by artificial intelligence is called Pascal VOC, and it is, to put it mildly, not very convenient.

Afterwards, the training of the artificial intelligence is re-initiated with the new resolutions of the images, and 50 000 steps should pass, which takes around 3-4 hours. Unfortunately, this practically doesn't increase the number of frames per second, and the artificial intelligence still lags on the graphics card.

Adding a second graphics card

Therefore, it was decided for an additional graphics card to be added. However, it is not necessary to add a new graphics card to the current computer. It is sufficient to connect another computer and code them to transmit images between each other using sockets [11]. For this purpose, a monitor, keyboard, and mouse need to be provided for the new computer.

In the end, the distribution of artificial intelligence and the game provided help. In the game, with the slowest TensorFlow and the Faster R-CNN model, it was possible to achieve 15 frames per second. And that is much better than the previous 5 frames per second. However, it is still an insufficient number for the artificial intelligence to aim properly. After all, in the game, there are at least 60 frames per second, while the artificial intelligence only sees 15 frames. This means that there are still 45 frames in the game where anything can happen. Most often, the opponent targeted by the artificial intelligence simply manages to move within those frames. Therefore, the speed of the current artificial intelligence is not sufficient, and something needs to be done about it.

YOLOv7

I mentioned that there are other architectures of neural networks because, obviously, TensorFlow Faster R-CNN is quite slow. Therefore, a decision was made to transition to a new artificial intelligence - YOLOv7. After a brief study of the artificial intelligence, YOLOv7 was downloaded. The solution is to conduct a short test, and as a result, everything works, and the machine intelligence quickly recognizes various objects in the test images. However, the code for capturing and processing images needs to be rewritten because it's no longer TensorFlow Faster R-CNN but PyTorch YOLOv7. On the first test of a YouTube video using only one graphics card without multithreading, YOLO achieves more than 25 frames per second. The situation in the game is not significantly worse. There, together with the game, it provides 20 frames per second. This is achieved with just one graphics card, the game running, and no optimization yet, and it already produces a sufficient number of frames for gameplay.

It's clear that the new artificial intelligence will need to be trained again. It will need to learn to distinguish opponents and teammates. The format for storing the points of the previous artificial intelligence was called Pascal VOC. In the case of YOLO, the format is different, and the existing XML files won't be of any help. They will need to be created from scratch. Fortunately, this task can be done with a converter rather than manually. After adjusting the configuration files of the artificial intelligence, the training process begins. Once the training is completed, the machine intelligence can be tested in practice, and everything works. It achieves around 20-25 frames per second and accurately identifies players. The only thing the artificial intelligence can't do on its own for now is target enemies, but this can be easily accomplished with auto-aiming code. After that, the artificial intelligence performs excellently in the game, considering that this is the most underdeveloped version, and the results are very good.

In addition to the registered result, there are also the conclusions of the report Improving Deep Object Detection Algorithms for Game Scenes [12], in which the use of five widely used object detection algorithms is analyzed, YOLOv3, SSD, Faster R-CNN, FPN and Efficient Det, and for eight games from various genres.

The contributions of this study are summarized as follows:

- A dataset of game scenes collected from eight games was built.
- A framework for improving the performance of object detection algorithms on game scenes was presented by retraining them using game scene datasets.
- Was tested whether the augmented images using image abstraction and stylization schemes can improve the performance of the object detection algorithms on game scenes.

The authors assert that retraining object detection algorithms using game scenes results in enhanced performance when compared to algorithms solely trained with public datasets like Pascal VOC and MS

COCO, which are commonly used in such studies. Their approach was rigorously tested, including the estimation of mean Average Precision (mAP) between pre-trained and retrained algorithms to demonstrate improved object recognition accuracy. Additionally, they estimated Intersection over Union (IoU) to showcase improved object localization accuracy. Furthermore, the authors conducted experiments with data augmentation techniques tailored for their specific purposes, revealing limited effectiveness based on the characteristics of the game scenes.

Conclusion

In conclusion, it can be said that artificial intelligence has great potential in many areas of life, and one of them is real-time object detection from displays. This type of artificial intelligence can be utilized in various industries and applications, such as the automotive industry, where it can be used for autonomous vehicle control, robotics, as well as in medicine and other fields.

Real-time object detection from displays is extremely valuable for numerous spheres of life and can be employed to address a wide range of problems. For example, this type of artificial intelligence can be used for rapid and efficient detection of issues in manufacturing processes, leading to significant improvements in production performance and cost reduction.

Just because many associate artificial intelligence with games does not mean we should overlook its vast potential to solve serious problems in different domains of life. And while artificial intelligence technology is still in its early stages, we are witnessing epochal changes it can bring in the future. Let us appreciate the significance of artificial intelligence and be excited about the future ahead, as its potential for transformation is limitless.

REFERENCES

[1] Wu, Y., A. Yi, C. Ma, L. Chen, Artificial intelligence for video game visualization, advancements, benefits and challenges, Mathematical Biosciences and Engineering, 20(8): 15345-15373, 21 July 2023

[2] Brandão, A., P. Pires, P. Georgieva, Reinforcement Learning and Neuroevolution in Flappy Bird Game. in Pattern Recognition and Image Analysis, eds.: A. Morales, J. Fierrez, J. Sánchez, B. Ribeiro, IbPRIA 2019. Lecture Notes in Computer Science, vol. 11867, Springer, Cham, 2019, pp. 225 – 236

[3] H. Su, A. Mariani, S. E. Ovur, A. Menciassi, G. Ferrigno, E. De Momi, Toward teaching by demonstration for robot-assisted minimally invasive surgery, IIEEE Transactions on Automation Science and Engineering., January 2021

[4] Hasan, T. H. R., A. Bibo Sallow, Face Detection and Recognition Using OpenCV. Journal of Soft Computing and Data Mining, Vol. 2(2), 2021, pp. 86–97.

[5] How To Take A Screenshot In Python Using MSS, 2022, https://nitratine.net/blog/post/how-to-take-a-screenshot-in-python-using-mss/

[6] Multithreading enables multiple threads of the same process to execute simultaneously (in Bulgarian), 2019, https://bg.gadget-info.com/difference-between-multitasking

[7] Jaiswal, A. Object Detection Using Haar Cascade: OpenCV, 2022, https://www.analyticsvidhya.com/blog/2022/04/object-detection-using-haar-cascade-opencv/

[8] Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, TensorFlow: A System for Large-Scale Machine Learning, Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), November 2–4, 2016, Savannah, GA, USA, 2016.

[9] Tensorflow: Extracting Image and Label from TFRecords File, 2023, https://saturncloud.io/blog/tensorflow-extracting-image-and-label-from-tfrecords-file/

[10] Chablani, M. YOLO — You only look once, real time object detection explained, 2017, https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006

[11] Socket programming, IBM, 2021, https://www.ibm.com/docs/en/i/7.1?topic=communications-socket-programming

[12] Jung M., H. Yang, K. Min, Improving Deep Object Detection Algorithms for Game Scenes, Electronics 2021, 10(20), 2527, https://doi.org/10.3390/electronics10202527

David Livshits *- student in Second English Language School "Thomas Jefferson", Sofia, with fields of interest in informatics, coding and information technologies.*

e-mail: david.d.livshits@2els.com

PhD Georgi Hristov *- teacher in Second English Language School "Thomas Jefferson", Sofia, with fields of interest in teaching, coding, information technologies and robotics.*

tel.: +359 894 00 49 44

e-mail: georgi.hristov@2els.com

Received on: 07.08.2023