AUTONOMOUS DRONE CONTROL FOR VISUAL SEARCH BASED ON DEEP REINFORCEMENT LEARNING

Uroš Dragović[1]*,
Marko Tanasković[1],
Miloš S. Stanković[1,2],
Aleksa Ćuk[1]

[1]Singidunum University,
Belgrade, Serbia

[2]Vlatacom Institute,
Belgrade, Serbia

Abstract:

Autonomous flight of drone using Deep Reinforcement Learning is an attractive area of research in recent years that gives excellent results. Autonomous drone flight is defined through a set of complex tasks for understanding the environment and navigating independently through it. Understanding the environment means that the drone knows its location in respect to other objects and that it can easily reach the desired location without collision. Extending the problem with a target search task increases the complexity and the necessity for using new tools and algorithms. In this paper, we present an approach in which a drone, in addition to learning to navigate in an unknown environment, learns how to find and approach an object a priori assigned to it as a target. In our approach, the drone uses RGB and RGB-D cameras as the only source of information about environment. Our proposed solution incorporates, into the framework of deep reinforcement learning, appropriate fast object detection, feature extraction, as well as efficient existing algorithms for avoiding obstacles. The proposed model uses the sensed RGB-D image of the drone as the main factor for estimating the distance to the obstacles, while, on the other hand, our model also requires two RGB images for a Siamese network as feature extractor used to identify the target in the environment, group of these images represents the current general state, based on which drone performs the action for which it can potentially receive the highest reward. We used a 3D simulator (MS AirSim) to validate the performance of our approach. Based on the simulation results, we conclude that the proposed method exhibits promising performance in terms of the rate of successful approach to the required target.

Keywords:

Deep reinforcement learning, drone, target search, computer vision, autonomous flight.

INTRODUCTION

In the last decade, there has been a rapid development in the field of artificial intelligence, especially machine learning algorithms. Advances and variations of multilayer neural networks, as well as great advances in computer technology, have allowed us to solve very complex and demanding problems of machine learning. In supervised learning and unsupervised learning, new approaches have yielded results that are close to, or in some cases, better than, human performance. For this type of learning, it is necessary to invest a lot of time in data collection and prep-

Correspondence:
Uroš Dragović

e-mail:
udragovic@singidunum.ac.rs

DOI: 10.15308/Sinteza-2022-382-388

aration in order for such algorithms to have exceptional results [1]. They tend to show the power they have and in which direction modern algorithms are going, but what will make a real difference in the future and change the way the world works are algorithms that can be trained without predefined data/instructions, the systems which can learn independently through behavior. Such algorithms belong to the field of reinforced learning [2].

Reinforcement learning algorithms play a major role in creating systems, agents or robots that can perform tasks independently, such as autonomous vehicles, factory plants, food delivery systems, and similar systems. These algorithms gave the first observed results in [3]. In the first approaches in which these algorithms were combined with deep neural networks, results were shown that give performance close to humans. But, by further improving the algorithms, human performance is far surpassed in some complex tasks such as the board game Go, in which the world champion in this game has no chance against the algorithm [4], or the video game Dota 2 in which there are a number of virtual characters that solve the problem by communicating with each other versus the team of 5 professional players who compete against them [5].

In this paper, we deal with solving the problem of autonomous drone search for a specific object in real world environment using a combination of the previously mentioned approaches. This problem was chosen because drones are already used today to solve search and rescue missions and inspect large plants or areas. The disadvantage of this use is that the person controls from a distance and in that way, it is difficult to coordinate the object in space, especially if a larger number of drones are involved. For this way of usage, people have to go through special training, but even after that, it is very difficult for them to move by drone through an unknown space because a person cannot have a completely clear picture of the environment while controlling remotely. If drones could understand and move through unknown space on their own and have the ability to identify the objects they see during the flight, then they would not need a man for direct control, and one person could be responsible for a larger number of drones.

Guided by the problem presented in this paper, we will present an algorithm that has good potential to solve this problem. The proposed solution uses Deep Q Learning as the base algorithm [3], whereas auxiliary, pre-trained supervised algorithms are also used to extract object features and localize target on the image. The problem of searching for an object is not only the identification of objects in space but also includes another set of complex problems, of which we list the two most important for us. The first problem is autonomous flight through the environment without colliding with other objects, whether static or dynamic. The second problem is the problem of localization in space. If a drone is not aware of the environment, it can endlessly repeat the search in the same area.

The localization and search can be successfully solved by using deep reinforcement learning. One approach was presented in [6], where the algorithm is always trained for a predetermined space. In [7], it has been shown that deep reinforcement learning can solve the problem of exploring an unfamiliar environment. Finding an object in an image can only be an initial task in one of the cases such as tracking a specific target as described in [8]. When objects are known in advance, the search for them can be facilitated by adding markers, or stickers with special visual characteristics, which is usually the case when locating landing sites, [9] and [10]. The searching process can also be defined through different types of recognition, such as estimating the position in [11], which can be used in systems with drones intended for surveillance. Based on the potential presented in the above cited algorithms, in our approach we use deep reinforcement training which is using information from two types of camera sources: color image and the corresponding depth map (distances to the objects in the image). Drone behavior is defined through discretization of the possible action values.

2. DRONE OBJECT SEARCH ALGORITHM

In order to find an object in an unknown dynamic environment, a number of problems must be solved. We will propose an algorithm and show that by using it, a drone can solve the search problem on its own, without human assistance, and without a combination of a number of complex algorithms. Our algorithm relies entirely on deep reinforcement learning with a combination of object detection and recognition algorithms.

For a drone that can autonomously move to find an object, the following indirect problems must be solved:

1. Avoiding obstacles in dynamic space
2. Recognition and identification of the required object
3. Localization of the drone in the environment

2.1. ALGORITHM ARCHITECTURE

The proposed algorithm's architecture is based entirely on deep Q learning. State-based Q value approximation is generated as a result of 3 tracks of neural networks. At the input to the algorithm, 3 images of the same dimensions 128x128 are given. The first part of the system is a network whose task is to approximate the distance and shapes of objects in the image. The other two images pass through network number 2, based on the ResNet50 architecture [12] which is used to extract the characteristics of objects in the image. The ResNet50 network was chosen because it gives one of the best results in the feature extraction process. One of those images is what the drone sees at a given moment and we get all the features from it. We repeat the same for the image on which the requested object is; however, unlike the first one, we temporarily store the features of this image in memory. The relationship between these two images is found through the new 3 layers of the neural network. After that, the approximation of the distance of the objects and the approximation of the features in the image are merged. At the output, there is a layer of 6 neurons for each of the possible actions. The architecture is shown in Fig. 1.

2.1.1. State

The algorithm is inspired by the way human searches for a certain object, in particular, by the information a person needs for searching. First of all, human needs to know what the object he is looking for looks like, for example, he needs to know visual characteristics such as shape, size, color. The next thing is to determine in which space we are performing the search. In order for a human to move through spaceit is primarily needed to use sight in order to avoid obstacles. Human visual sensing is typically based on two receivers, two eyes, which makes it possible to get the information about shape and color with the same signal, and also to estimate the distance from the objects. The last thing that is necessary is localization in space and environment mapping; for instance, a person does not want to look twice in the same place when searching. In relation to this explanation, we can define what we need during the search process. Since this algorithm is based on the algorithm of deep Q learning, we need to appropriately define states, actions and rewards. The algorithm expects 3 components of the input state:

1. Target image - an image in the red-green-blue (RGB) spectrum that clearly shows the desired object and occupies the surface of almost the entire image, this image should be in 128x128 format. This image is used to extract the main features of the desired object
2. Image from the drone camera - we assume that the drone is equipped with a monocular camera, from which it gets a real-time image in the red-green-blue (RGB) spectrum, the image should be 128x128. This input signal primarily serves us to identify the characteristics of the required object if the drone is aimed at it.

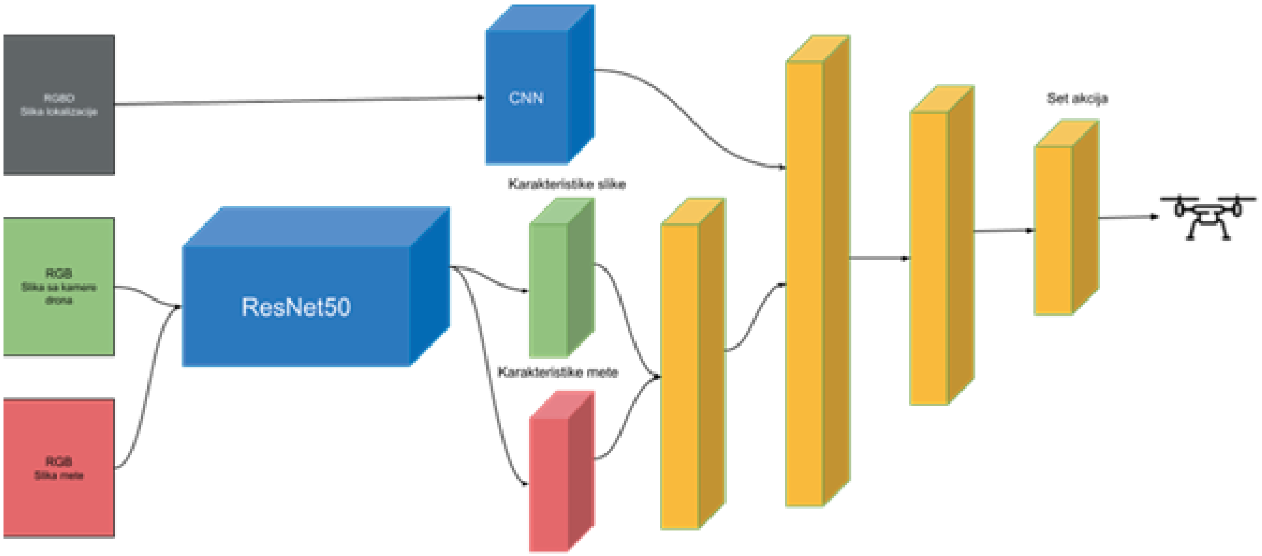

Figure 1 - Network architecture

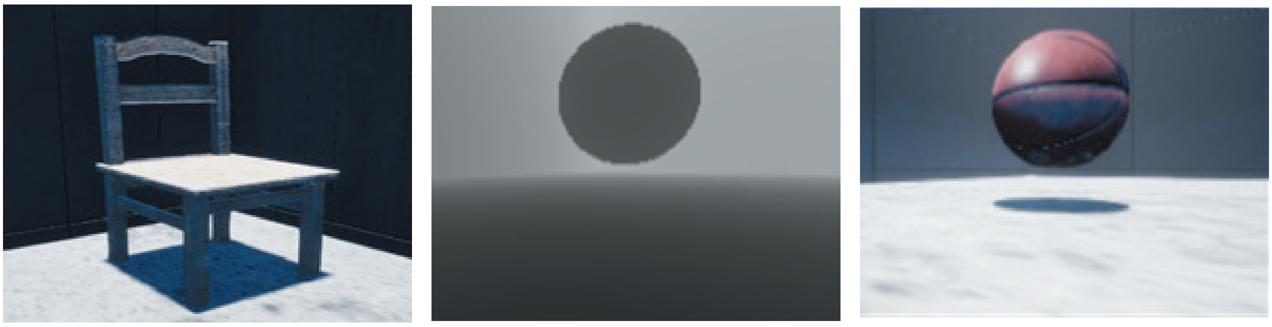

Figure 2 - Example of 3 input images, target (left), RGB-D (middle), RGB (right)

3. Depth camera image - it is assumed that the drone is equipped with an additional, RGB-D camera which, in addition to the classic color image, determines the depth in space, i.e. how far the objects in the image are. In the algorithm, this input signal is primarily intended to contribute to the easier avoidance of obstacles.

These three components provide enough information to be able to define the state of the drone, taking into account the target, the type of drone, and the relative localization of objects in space.

2.1.2. Action

A drone is an object that can move freely in all 3 axes x, y and z; it does not have a defined front because it can move in all directions equally. For this algorithm, the front of the drone is the side on which the camera is pointed forward. The set of actions that a drone can perform are up, down, forward, turn left and turn right for 30°, as well as stopping.

As can be seen from the sequence there are 6 possible actions, movement up and down is determined by moving at a constant speed for a given constant time in one of these two directions. A 30 ° rotation of the angle was chosen so that the drone could determine the visual movement of the characteristics of the objects in the image when moving. In order for the drone to turn in the opposite direction, it is necessary to perform 6 left or right turns. Although the drone has the ability to move backward relative to the camera, this is not provided by this algorithm.

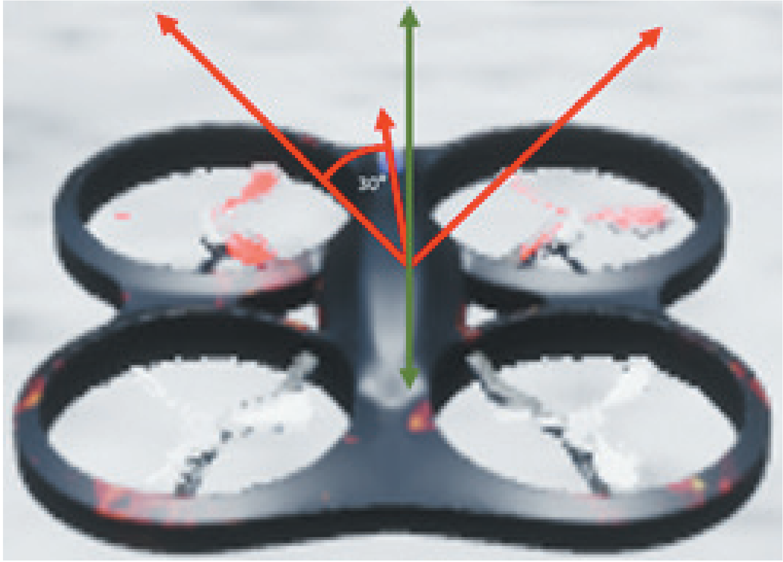

Figure 3 - Image describes directions in which it can moves

2.1.3. Reward

After the described conditions and actions, we can define the way of awarding prizes for the behavior of the autonomous drone.

To prevent a drone from stopping for a long time, each time the stop action is chosen 2 times or more in a row the reward the agent received is negative, amount of -0.2, and each time in a series of stop choices the reward value is reduced twice as much as before.

$$f(x) = \begin{cases} 0, & x < 2 \\ -0.1 * x, & x \geq 2 \end{cases}$$

Where x is a number that determines how many times in a row the stop action is selected.

When the drone hits one of the objects in space, the execution of the episode is stopped immediately, and the drone receives a negative reward of -100.

In order to receive a positive reward, a drone must have in its field of vision the required object or object that looks like it more than 70%. Estimation of similarity is determined by running an image of the object through the described feature extraction algorithm (ResNet50) and its approximate values are stored throughout the episode. The input RGB image from the drone, which is forwarded to the input as the state of the algorithm, is also passed through the YOLO [13] algorithm, after detecting all objects in the image for each object, the feature values are determined using the ResNet50 algorithm. The estimation of similarity for each object in the picture in relation to the required object is determined by the Euclidean distance. Then, for each object that has a similarity of more than 70%, it is determined which surface it occupies in the picture, which is proportional to the distance of the drone from the object. The closer the drone is, the larger the object occupies, so the final prize is calculated $r = \frac{p}{10} * 2$, where p is the area object covers in percentage.

Acceleration of agent search is achieved by giving a small negative reward when there is no object in sight with a similarity greater than 70%.

$$r_t = -0.05$$

2.2. LEARNING

The learning process is performed using the deep Q learning scheme, with the above defined variables. The exploration/exploitation strategy is based on the ε-greedy policy, with the exploration probability ε being reduced by a small step after each completed episode. For good generalization, the originally proposed architecture with two networks is used: the final neural network and the training Q neural network. The use of a standard optimizer and the use of a system to replicate the experience gained is also retained. Neural networks responsible for extracting features from the image and localization of objects in the image are used in their original form, with the parameters with which they give the best results.

3. EXPERIMENT AND RESULTS

The proposed algorithm was trained and tested in the AirSim simulator [13] ,which is open source and is intended primarily for researchers in the field of artificial intelligence. The Python programming language with the Torch library, deep neural networks framework, was used to implement the algorithm.

In the simulator, we created a simple training environment surrounded by walls to limit the space the drone searches. No physical element has been placed on the upper side of the environment to limit the movement of the drone, but the upper limit is conditionally limited to 20 units. When the drone went out of this frame, it was considered as it hit one of the objects. The shape of the environment can be seen in Figure 4.

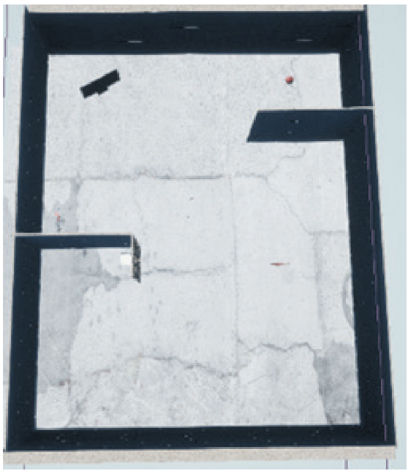

Figure 4 – Training environment

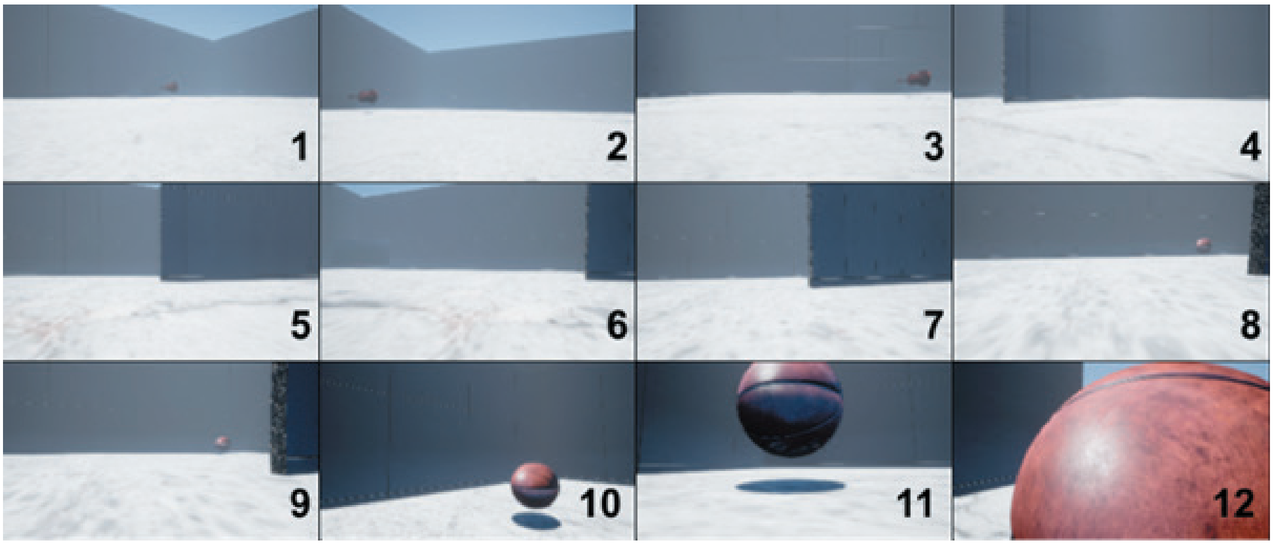

Figure 5 – An example of selected frames from drone in one episode

The simulator contains 4 items that can be searched for: a chair, a TV, a ball and a bicycle. These 4 objects were not chosen at random, they were chosen because the previously presented YOLO algorithm has available weights that give reliably good results with these objects. Photographs of each of the objects were created before the start of the training, so that it would be possible to specify the image on which the target is located. Each of the elements in each new episode is placed in a random position within the walls. Every element is always included in the environment whether it is sought after or not. In this way we get a higher degree of generalization for parameters.

After a training process that lasted more than 40 hours on standard PC with Intel i7 CPU, 1660Ti GPU supported with 32GB of RAM, we determined the success measure according to whether the drone successfully performed the task for which it was trained. In the following image, we can see 12 selected photos generated in the moment of searching for the ball, where the photo with number 1 is the beginning of one episode, and the photo with number 12 is the end of that episode, photos in between are taken at random moments in the given order.

5. CONCLUSION

In this paper, we presented a new object search algorithm by an autonomous drone using only visual and depth inputs, based on deep reinforcement learning, together with deep learning-based object detection schemes. With the presented algorithm, we have shown that the visual information obtained from the drone camera can be used efficiently, similar to eyes used by humans and animals. The fact that the drone "independently" overcame the problem of finding the required object shows how much potential and effectiveness is hidden in the reinforcement training algorithms. With this work, we have shown that deep neural networks can be used in the process of determining rewards, and not only as approximators of the Q table of values.

The potential further development of this research can go in several directions. One is the enhanced approach to deep Q training. In the presented work, the actions are discrete values with precisely defined drone displacement. A possible generalization is to use continuous values instead of discrete ones, for example, speed in all three directions, which would enable the drone to move much more precisely, but also much more aggressively if necessary. With this improvement of our algorithm, we would bring its architecture closer to the deep deterministic gradient algorithm. In addition to improvements in the way they move, there is potential for improving the detection of the desired object. For example, an introduction of the division of parameters at multiple levels of the neural network when extracting characteristics would strengthen the link between what the drone sees and what it seeks. In order to improve the search speed, we could extend the algorithm to work in a decentralized multi-drone setting.

6. ACKNOWLEDGEMENTS

This research was supported by the Science Fund of the Republic of Serbia, Grant #6524745, AI-DECIDE.

7. REFERENCES

[1] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, p. 436–444, 2015.

[2] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, Cambridge, MA, USA: A Bradford Book, 2018.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," 2013.

[4] D. Silver, A. Huang, C. Maddison and e. al., Mastering the game of Go with deep neural networks and tree search, Nature, 2016.

[5] OpenAI, *OpenAI Five*, https://blog.openai.com/openai-five/, 2018.

[6] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei Fei and A. Farhadi, "Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement," CoRR, 2016.

[7] A. Peake, J. McCalmon, Y. Zhang, D. Myers, S. Alqahtani and P. Pauca, "Deep Reinforcement Learning for Adaptive Exploration of Unknown Environments," *CoRR*, 2021.

[8] H. Zhang, G. Wang, Z. Lei and J.-N. Hwang, "Eye in the Sky: Drone-Based Object Tracking and 3D Localization," *CoRR*, 2019.

[9] M. Abdelkader, T. Noureddine, U. A. Fiaz, N. Toumi, M. A. Mabrok and J. S. Shamma, "RISCuer: A Reliable Multi-UAV Search and Rescue Testbed," *CoRR*, 2020.

[10] M. Beul, S. Houben, M. Nieuwenhuisen and S. Behnke, "Fast autonomous landing on a moving target at MBZIRC," *European Conference on Mobile Robots (ECMR)*, pp. 1-6, 2017.

[11] S. Penmetsa, M. Fatima, S. Amarjot and O. S. N, "Autonomous UAV for suspicious action detection using pictorial human pose estimation and classification," *ELCVIA: electronic letters on computer vision and image analysis*, pp. 18-32, 2014.

[12] K. He, X. Zhang, S. Ren and J. Sun, *Deep Residual Learning for Image Recognition*, 2015.

[13] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, *You Only Look Once: Unified, Real-Time Object Detection*, 2015.

[14] S. Shah, D. Dey, C. Lovett and A. Kapoor, "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," in *Field and Service Robotics*, 2017.