ICIC Express Letters
Part B: Applications
Volume **13**, Number **2**, February **2022**

ICIC International ©2022 ISSN 2185-2766
pp. **195–202**

DEEP Q-NETWORK CONFIGURATION AND PERFORMANCE FOR A POWER LINE INSPECTION AUTONOMOUS QUADROTOR IN A SIMULATED WORLD

Muhammad Idham Ananta Timur*, Jazi Eko Istiyanto, Andi Dharmawan
Rois Nur Hakim and Ahmad Shiddiq

Department of Computer Science and Electronics
Universitas Gadjah Mada
Sekip Utara, BLS 21 Yogyakarta 55281, Indonesia
*Corresponding author: idham@ugm.ac.id
{ jazi; andi_dharmawan }@ugm.ac.id; { rois.nur.hakim; a.shiddiq }@mail.ugm.ac.id

Received March 2021; accepted June 2021

Abstract. *Monitoring power line across mountainous landscape is always a daunting task. Usually, pilot-driven unmanned aerial vehicle (UAV) such as quadrotor is one of the most viable options to do so. However, this scenario puts a high demand on the pilot both in availability and costs. Therefore, this study attempts to adopt autonomous approach using Deep Q-Network algorithms to maintain UAV position against the power line. Sum of rewards per episode and loss function analysis are used for the evaluation. The result shows that adjusting number of actions and kernel size could significantly accelerate learning process while maintaining an acceptable sum of rewards. On the other side, bigger kernel size and more actions will slow the learning process down but gain even better sum of rewards.*
Keywords: Deep Q-Network, UAV, Reinforcement learning, Power line monitoring

1. **Introduction.** Safety is an important concern in power transmission industry. One of the biggest challenges is in monitoring the power line across rural, inhabited, and mountainous landscape. Human patrol is costly and inefficient, not to mention in some cases leading to both dangerous and unhealthy experience. Moreover, recent business pressure requires real-time demand in monitoring and maintenance.

One of the prospective options is an autonomous UAV (unmanned aerial vehicle). A UAV provides a view from the top, which usually has a better vision as it has less obstruction. However, manually driven UAV also needs a pilot who should be close enough to control the UAV. This human involvement has the same risk with human patrol for his/her health and safety especially in extreme weather or circumstances such as natural disaster. Thus, autonomous UAV which is able to follow power line with much less human intervention will be a big help [1].

Modern UAV has a decent in-flight control system. Nonetheless, in a monitoring task which heavily relies on camera position, GPS and IMU sensor might not be enough due to their lack of accuracy and intelligence. One control system that is often implemented is proportional integral derivative (PID) control [2]. It cannot cope with dynamic conditions such as wind, variable loads, and voltage drops, resulting in suboptimal system response [3]. Therefore, it is necessary to develop a smart system that can perceive image from its camera and intelligence to adjust the conditions and circumstances so that the quadrotor can do its job properly.

There are several studies that investigate a more efficient way to do autonomous power line inspection. Vega et al. [1] discussed an inspection of high-voltage power transmission

DOI: 10.24507/icicelb.13.02.195

lines using a quadrotor helicopter. Inspection of the transmission line is carried out using manual flight controls with a color camera and a TIR camera to provide information on locating overheated devices and components. Knowing this state can determine some of the common faults that exist in the power cable. Teng et al. [4] proposed a solution for the LIDAR mini-UAV electric cable inspection system using the AOEagle system using manual flight controls. The result shows that the performance of the AOEagle system has been well verified, especially for mountainous environments with minimal costs.

Route planning autonomous technology could also improve efficiency. It starts with manually operated waypoints with the camera position and angle of each waypoint recorded through flight control planning function. Next, those waypoints are connected, and the automatic detailed inspection is generated automatically [5]. While this approach achieved better inspection efficiency and position accuracy rather than traditional alternatives, it still employs a good amount of manual operation along the way.

Higher level of UAV autonomy could be reached by adopting intelligent systems both in object detection and UAV controls. Supervised learning is the popular choice for object detection. A vision-based autonomous navigation approach to inspecting transmission lines using an unmanned aerial vehicle (UAV) has been proposed [5]. The process of detecting transmission lines is based on a region-based convolutional neural network (R-CNN). Then the results of this study inform that continuous flight without GPS shows the effectiveness for the navigation model and the transmission line inspection model. In another case mask regional convolutional neural network (Mask RCNN) has been deployed on UAV for power line detector. Using deep learning Resnet50 architecture and feature pyramid network (FPN) architecture for feature extraction on a pile of UAV's camera images, this approach has successfully detected insulator, transformer, and power pole on a power line [6]. In another case, improved RetinaNet and Cascade RCNN structure has been used for object detection in aircraft surface inspection with good accuracy and detection speed [7]. Object detection however is only part of the process, as how to autonomously fly the UAV is another significant step that must be addressed to achieve autonomous solution.

The ability to map from current state perceived by a UAV into optimal actions/controls after training is where the Deep Q-Network shines. However, the challenge is working with big number of data, because the amount of training data required to study a state into action is quite large. It can be a problem in a real world as the autonomous vehicles are often unsafe to operate during the training phase. Using a simulation instead can reduce the risk and cost if something goes wrong. The use of simulations to fly a quadrotor following a power cable with various states will efficiently display a better understanding of the method being applied, without having to risk any safety measures.

In this paper we study an autonomous approach of UAV power line inspection by exploring the configuration and performance of DQN in a simulated world. The result of this study shows that a proper configuration DQN shows a promising performance in learning how to fly the UAV to accomplish the task. The rest of the paper is structured as follows. Section 2 addresses a few of similar works in this area. Section 3 describes the simulator, training architecture and DQN parameters configuration. Section 4 discusses the experimental results and performance analysis. Section 5 concludes the paper.

2. **Relevant Works.**

Similar work [8] is used to determine the performance and accuracy of the inner control loop when providing attitude control when using an intelligent flight control system trained on a quadrotor using a reinforcement learning algorithm, trust region policy optimization (TRPO), and proximal policy optimization (PPO). Then this study also conducted performance comparisons between PID controllers to identify whether using the reinforcement learning method was appropriate for high-precision and time-critical flight control. The result of this research is that the reinforcement learning

method can train accurate attitude controllers. Later in training with PPO this method was able to outperform a fully tuned PID controller on almost every metric.

Another research has been carried out in the simulated reinforcement learning method in simulator and its trustworthy autonomy [9]. This study describes the steps required to create a drone simulation environment suitable for experimenting with vision-based reinforcement learning. It trains drone to navigate and collect cubes in a large environment. In addition, it has shown how to use existing deep neural network visualization techniques to understand the control policies that are generated when learning is running well.

3. **DQN Configuration for Autonomous UAV.** In this section we will discuss the configuration and implementation of Deep Q-Network for our autonomous quadrotor which is loosely based on [10].

3.1. **AirSim setup and configuration.** This work is built on AirSim [10], an unreal engine based open-source simulator for drones, cars, and more. It allows data collection to train those autonomous vehicles without having to use a real one. Each vehicle in AirSim could be manually controlled either by keyboard, joystick, or steering wheels, or autonomously trained using its application programming interface (API). Vehicle physical attributes and control such as location, steering angle, throttle, and distance could also be recorded during training phase. All these features make it possible to completely simulate and train a reinforcement learning UAV.

AirSim has a record feature to capture all images from UAV's front camera. In this work we also use its sensors to determine its speed and position. UAV's relative distance from the power line is calculated using Euclidean distance from a point (UAV coordinates) perpendicular to a line drawn between a tower and the next tower coordinates. All UAV's movements are programmatically controlled using python API. Fortunately, there are a few photo-realistic environments available which are suitable to emulate UAV power line inspection. One of them is mountain landscape, as shown in Figure 1, which has a long power line over mountainous land and snowy place and will be used in this work.

FIGURE 1. Virtual environment for power line monitoring

3.2. **Deep network architecture.** Deep Q-Network [11] is the beginning of next generation reinforcement learning research. It introduced the concept of replay buffer, which is inspired by biology mechanism called experience replay. It allowed significant data efficiency as each step can be reused to learn the Q-function. Mini batch samples could also be consecutively collected whenever replay buffer is not available to increase the variance of the updates, as those batches are highly correlated. Furthermore, experience replay

can smooth out learning since it avoids the situation when samples used to train are determined by the precedence parameters.

Deep Q-Network also unveils the idea of target network to generate Q-learning target to improve the stability of neural networks. It is periodically synchronized with the main Q-network either by copying directly or exponentially decaying average as shown in Figure 2. As a result, it reduces the divergence and oscillations because of the delay of their Q-learning target generation with old parameters.

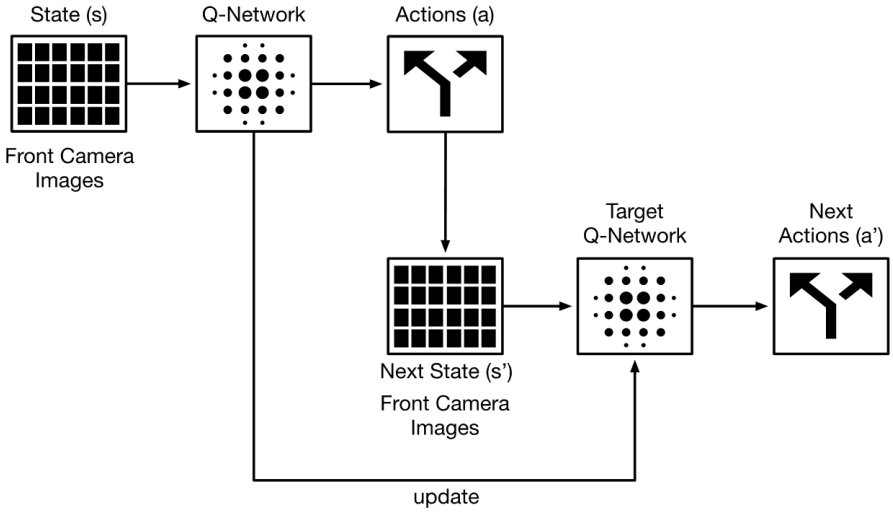

FIGURE 2. Learning network and target network

In this research, there are several ways that will be done to parameterize Q using a convolutional neural network. The convolutional neural network model uses several convolutional layers with max polling, dropout, fully connected, and activation functions [9]. The architectural model in this study is a hidden layer or known as the first hidden layer that combines 16 filters from 8×8 with stride 4 with the input image and applies non-linearity. The second hidden layer will work by combining 32 filters from 4×4 with a stride 2, and with a non-linear rectifier. Then the third hidden layer will combine 32 filters from 3×3 with stride 1. Furthermore, the last hidden layer will be fully connected and consists of 256 rectifier units. The output layer is a completely linear layer with a single output for each available action.

3.3. **Reinforcement learning configuration for power line inspection UAV.** State contains the current state information that will be used in determining the next action. The state in this study was obtained from quadrotor front camera. In addition to the image used, the current quadrotor position, speed and collision status during flight are read from the sensor provided in the simulator. The state at time t, denoted by s_t, is obtained by combining this information.

Preprocessing is used as an initial step in processing image from input data before entering the main stages of the DQN algorithm. At this stage, the image obtained by the quadrotor during the flight measuring 256×144 pixels will be converted into an image measuring 84×84 pixels. The steps that will be carried out in preprocessing are started by taking an image frame and then created in the form of an array data information containing the float data type. Information that has been processed into array data takes the maximum value for each pixel color value above the frame encoded with the previous frame. The second step is to extract the Y channel known as luminance from the RGB frame and change the shape to a scale of 84×84 pixels. The function Φ of the DQN algorithm is used to apply preprocessing to the most recent frame and stack it to produce input to the Q function.

Action is a form of interaction that occurs between the agent and the environment. The actions given will be based on the flight speed setting of the UAV which is on the x, y and z coordinate axes. The coordinate system provided in the AirSim simulator uses the NED (North, East, Down) coordinate system. The types of actions provided include hovering action, forward flying action, forward flying and turning right action, forward and downward flying action and flying forward and turning left. Each action performed at time t, will be denoted by a_t. Each action selected at time t will be given a reward.

The reward is a sum of distance reward and speed reward. Distance reward is inversely proportional distance between UAV and the power line, and speed reward is a proportional speed on x, y and z axes. UAV will reset its position if it travels further than certain threshold or crashes/hits another object. There is a penalty for every crash.

The training in this research will evaluate the value of the reward obtained. The training is made using experience replay techniques to train the Q-network. In training, the experience of each episode needs to be saved [12]. So in this research, replay memory is used to store the transition and agent experience during training at each s_t, at, s_{t+1}, reward, and done. Replay memory uses gradient descent minibatch which is a variation of the gradient descent algorithm that divides a training dataset into small batches that are used to calculate model errors and update model coefficients. This minimal gradient descent tries to find a balance between the robustness of the stochastic gradient derivative and the efficiency of the batch gradient descent. Then for behavioural policies during the training phase ε-greedy is used to conduct exploration. Behavioural policy is important to use because it determines the speed of convergence [13]. The behaviour policy during the training phase using the ε-greedy is set with a value range of 1.0 to 0.1 with over the first 1,000,000 frames and stays at 0.1 thereafter. The exploration policy using ε-greedy will be used as the basis for randomly selecting actions from the available actions. The complete algorithm for DQN is shown in Figure 3.

Algorithm: Deep Q-Network

Initialize replay memory (D) with the capacity (N)
Initialize action-value (Q) with random weights (θ)
Initialize target action-value $\left(\hat{Q}\right)$ with weights ($\theta^{-} = \theta$)
for episode = 1, M do
    Initialize sequence of $s_1 = \{x_1\}$ and preprocess sequence $\Phi_1 = \Phi(s_1)$
    for t = 1, T do
        with probability ε select random action (a_t),
        otherwise select $a_t = \arg\max_a Q(\Phi(s_t), a; \theta)$
        execute action a_t on AirSim and observe reward r_t and images x_{t+1}
        set $s_{t+1} = s_t$, a_t, x_{t+1} and preprocess $\Phi_t = \Phi(s_{t+1})$
        store transition (Φ_t, a_t, r_t, Φ_{t+1}) in D
        sample random minibatch of transition (φ_j, a_j, r_j, θ_{j+1}) from D
        If $D_i \neq 0$ set $y_j = r_j$ otherwise, $y_j = r_j + Y \max_{a'} \hat{Q}(\varphi_{j+1}, a'; \theta)$
        perform gradient descent step
        synchronize the target $\hat{Q} = Q$ in every C step
        if the episode has ended, break the loop
    end for
end for

FIGURE 3. DQN algorithm

4. **Experimental Result and Discussion.** The application of the Deep Q-Network algorithm for quadrotor testing was carried out for six flight tests using several parameters that have been made. The provisions made are based on several considerations. One of the considerations is the size of convolutional filters used in the test, the second is the reward systems, and the third one is the number of types of action provided. The results of the quadrotor flight test flown with these conditions are displayed in Figure 4. The graphs obtained consist of a loss function graph and a graph of sum rewards per episode. The loss function graph will show the accuracy of selecting the action taken by the agent and the graph of the sum rewards per episode will show evidence of the success of the learning carried out by the system.

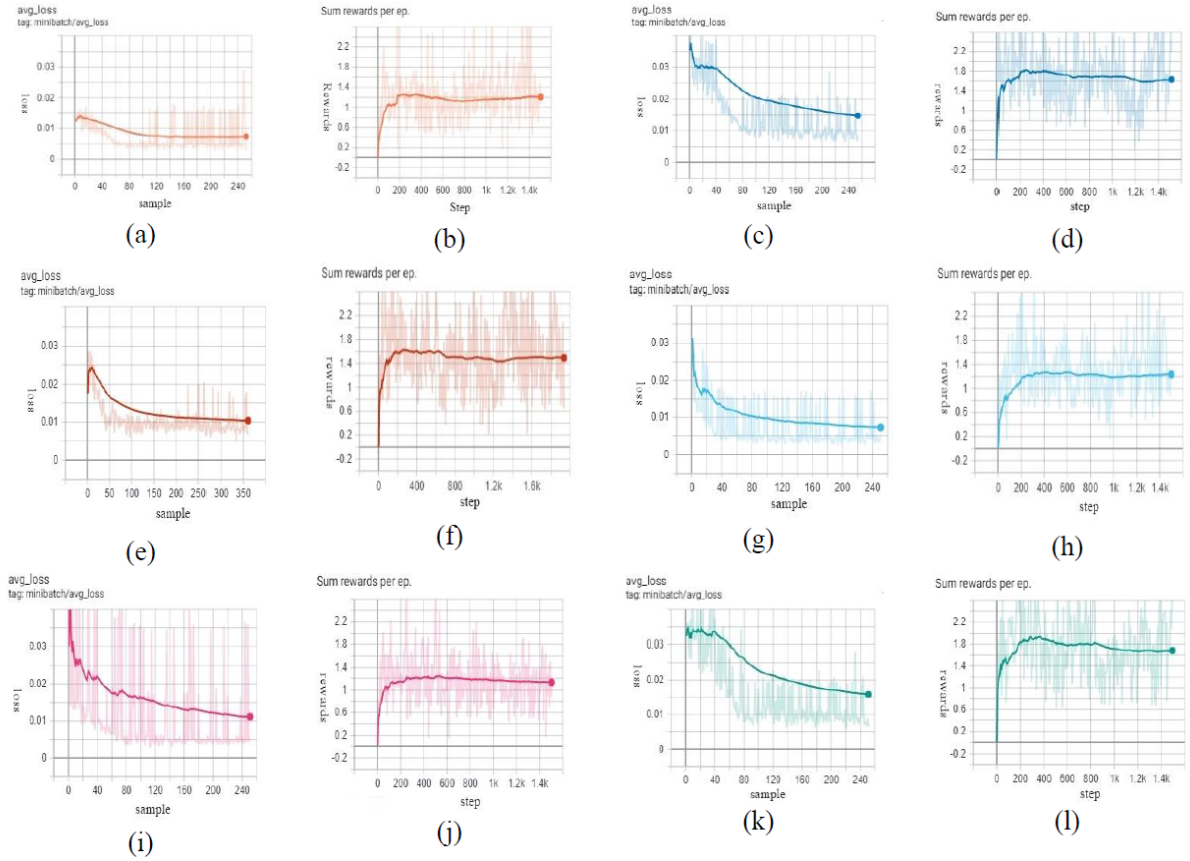

FIGURE 4. Experimental result: (a) (b) 3 rewards, 16, 32, 32 kernel, 5 actions; (c) (d) 2 rewards, 16, 32, 32 kernel, 5 actions; (e) (f) 2 rewards, 32, 32, 64 kernel, 5 actions; (g) (h) 3 rewards, 32, 32, 64 kernel, 5 actions; (i) (j) 3 rewards, 16, 32, 32 kernel, 6 actions; (k) (l) 3 rewards, 32, 32, 64 kernel, 6 actions

Based on the test results obtained as Figure 4, the loss function value of each test has decreased and tends to converge with a small value and is close to zero. The decrease in loss value which tends to be close to zero and the graph converges. This shows that the system has been running well and the selection of actions taken by the agent is getting more precise from one step to the next. Then on the graph of sum rewards per episode, the rewards value obtained is greater and tends to converge. The value of sum rewards per episode obtained shows that the learning system to maximize the rewards value has gone well and optimal. In addition, the results of this study also show that during the quadrotor flight process using the reinforcement learning method and the Deep Q-Network algorithm the system does not experience collisions. This statement is based on the results seen that the value of sum rewards per episode obtained is always positive.

Figure 4 shows tests using three reward functions, testing using convolution filters 32, 32, 64 and testing using convolutional filters 16, 32, 32 values tend to be the same or with not much difference. As expected, the use of convolutional filters 32, 32, 64 demands higher computation process and takes longer time than the use of convolutional filters 16, 32, 32. Figure 4 also shows a higher reward value for 2 rewards systems compared to 3 rewards. One missing reward in those systems is the distance function to the nearest next tower location. In practice, 2 rewards system successfully maintains quadrotor distance against the power line. However, most of the time it only hovers around the power line center. Using 3 rewards systems could 'motivate' the UAV to go to the next destination. This shows unique feature of reinforcement learning where it is able to autonomously learn environments dynamics based on goal. It differs from more conservative supervised learning where the model was like an unexplainable black box which needs to be retrained whenever there was a significant change in the environment.

Figures 4(i), 4(j), 4(k) and 4(l) put additional action which is slower moving forward into the quadrotor. Interestingly, it achieves higher rewards when using bigger convolutional filters. It shows that slowing down the movement and putting more detail of the kernel will get a better learning experience though it is a little bit slower to converge.

5. **Conclusion and Future Works.** This work shows the feasibility of DQN algorithm to be implemented in autonomous quadrotor power line inspection as it shows convergence in a fairly short time, although the results still vary. Moreover, reducing number of actions and less kernel size could significantly accelerate learning process while maintaining an acceptable sum of rewards. On the other side, bigger kernel size and more actions will slow the learning process down but gain even better sum of rewards. In addition, the use of the addition of the reward function regarding the distance between the quadrotor and the point of interest in this study can further motivate the quadrotor to move and achieve convergence.

Based on the research that has been done, there are suggestions that can be developed for further research, including modification of the reward model for the same flight mission to optimize the level of system convergence. Then a broader implementation can be done for the use of the reinforcement learning method. In other words, it can develop a reinforcement learning method using the Deep Q-Network algorithm as a navigation flight or something else. Then the second is that the reinforcement learning method with the Deep Q-Network algorithm can be compared its performance with other algorithms.

Acknowledgment. This work is partially supported by Final Project Recognition Grants from Ministry of Education of Republic of Indonesia.

REFERENCES

[1] L. F. L. Vega, B. Castillo-Toledo, A. Loukianov and L. E. Gonzalez-Jimenez, Power line inspection via an unmanned aerial system based on the quadrotor helicopter, *MELECON 2014 – 2014 17th IEEE Mediterranean Electrotechnical Conference*, Beirut, Lebanon, pp.393-397, 2014.

[2] W. Koch, R. Mancuso, R. West and A. Bestavros, Reinforcement learning for UAV attitude control, *arXiv.org*, arXiv: 1804.04154, 2018.

[3] K. N. Maleki, K. Ashenayi, L. R. Hook, J. G. Fuller and N. Hutchins, A reliable system design for nondeterministic adaptive controllers in small UAV autopilots, *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, Sacramento, CA, USA, pp.1-5, 2016.

[4] G. E. Teng, M. Zhou, C. R. Li, H. H. Wu, W. Li, F. R. Meng, C. C. Zhou and L. Ma, Mini-UAV LIDAR for power line inspection, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLII-2/W7*, pp.297-300, 2017.

[5] T. He, Y. Zeng and Z. Hu, Research of multi-rotor UAVs detailed autonomous inspection technology of transmission line based on route planning, *IEEE Access*, vol.7, pp.114955-114965, 2019.

[6] S. Vemula and M. Frye, Mask R-CNN powerline detector: A deep learning approach with applications to a UAV, *AIAA/IEEE 39th Digital Avionic Systems Conference (DASC)*, 2020.

[7] B. He, B. Huang, Y. Lin and L. Wu, Intelligent unmanned aerial vehicle (UAV) system for aircraft surface inspection, *The 7th International Forum on Electrical Engineering and Automation (IFEEA)*, 2020.

[8] J. Luo, S. Green, P. Feghali, G. Legrady and Ç. K. Koç, Reinforcement learning and trustworthy autonomy, in *Cyber-Physical Systems Security*, Ç. K. Koç (ed.), Cham, Springer International Publishing, 2018.

[9] X. Hui, J. Bian, X. Zhao and M. Tan, Vision-based autonomous navigation approach for unmanned aerial vehicle transmission-line inspection, *International Journal of Advanced Robotic Systems*, vol.15, 2018.

[10] S. Shah, D. Dey, C. Lovett and A. Kapoor, AirSim: High-fidelity visual and physical simulation for autonomous vehicles, *arXiv.org*, arXiv: 1705.05065, 2017.

[11] V. Mnih, K. Kavukcuoglu, D. Silver et al., Human-level control through deep reinforcement learning, *Nature*, vol.518, no.7540, pp.529-533, 2015.

[12] M. Andrychowicz et al., Hindsight experience replay, *Proc. of the 31st International Conference on Neural Information Processing Systems*, Long Beach, CA, USA, pp.5055-5065, 2017.

[13] S. Wender, *Integrating Reinforcement Learning into Strategy Games*, Master Thesis, University of Auckland, 2009.