Visual Reinforcement Learning for Dynamic Object Detection

Xiangsheng Wang [1], Xikun Hu [1], Ping Zhong [1]

[1] College of Electrical Science and Technology, National University of Defense Technology, China -
(wangxiangsheng, huxikun, zhongping)@nudt.edu.cn

Keywords: Reinforcement Learning, Dynamic Object Detection, Viewpoint Adjustment.

Abstract

Object detection is a widely studied task in computer vision. Current methods often focus on images captured from appropriate viewpoints. However, there is a large disparity between objects observed from different viewpoints in the real world. Dynamic Object Detection (DOD) method automatically adjusts the camera viewpoint in a visual scene to sequentially find optimal viewpoints. Currently, the DOD tasks are usually modeled as a sequential decision-making problem and solved using reinforcement learning methods. Existing approaches face challenges with sparse rewards and training instability. To tackle these issues, we proposed a single-step reward function and a lightweight network, respectively. The single-step reward function, which provides timely feedback, gives an efficient training process for DOD tasks. The lightweight network with few parameters can ensure the stability of the training process. To evaluate the effectiveness of our method, we developed a simulation dataset based on UE4, which consists of 1800 training images and 450 testing images. The dataset includes five object categories: vans, cars, trailers, box trucks and SUVs. Experiments demonstrate that our method outperforms SOTA object detectors on our simulation dataset. Specifically, the average precisions(APs) are improved from 89.1% to 96.0% when using the YOLOv8 object detector.

1. Introduction

Visual object detection is widely used in a variety of industries. In recent years, significant progress has been made in the field of object detection with the development of deep learning. Existing object detection methods, such as Faster R-CNN (Ren et al., 2017) and YOLO (Redmon et al., 2016), have achieved satisfactory detection performance. Object detection technology has shown promising applications in various fields such as traffic monitoring (Byun et al., 2021), power inspection (Abdelfattah et al., 2020) , and disaster relief (Boi-Tuli et al., 2019).

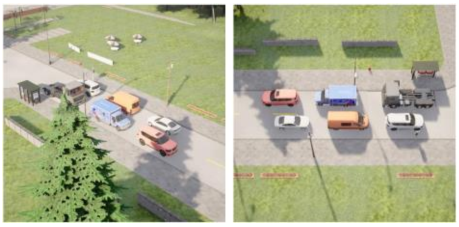


Figure 1 Images captured from two different viewpoints

Current methods often focus on images captured from appropriate viewpoints. However, real-world applications often involve images taken with uncertain intentions, small-scale objects, and partial occlusions, which can result in poor detection performance. As shown in Figure 1, objects are partially obscured in the left image, which can be solved by adjusting the viewpoint as shown in the right image.

Figure 2 presents the objects imaging under 5 different yaw angles(0°, 45°, 90°, 135°, 180°) and 3 different pitch angles(90°, 60°, 30°). It can be noticed in Figure 2 that there are some views where objects are heavily occluded from each other, and there are also some views where targets are not occluded from each other. For the same objects in the scene, imaging with different viewpoints will get a different detection score. To improve detection performance in real-world scenarios, it is of vital importance to find a proper viewpoint.

Dynamic Object Detection(DOD) method, which automatically adjusts viewpoints in a visual scene to sequentially find optimal viewpoint and scale, can effectively circumvent poor viewpoints and achieve better detection performance. There is an object detector and a next-view selector in a DOD framework. The object detector provides the object detection result towards the current image, and the next-view selector adjusts the viewpoint based on the detection results to achieve a better detection score in the next image.

As for training, a two-stage training framework has been designed for DOD methods (Xu et al., 2021; Ding et al., 2023). The first stage aims to obtain a well-trained SOTA object detector, such as YOLOv8 or other similar detectors. The subsequent stage is to train an effective next-view selector by receiving feedback from the object detector. Sequentially adjusting the viewpoint and scale can considerably enhance the detection performance of the object detector for objects of interest in the following observations.

The adjustment process in DOD problems is usually modelled as a sequential decision-making problem and solved using reinforcement learning methods. Current DOD methods typically design episode reward(Han et al., 2019) and extract image features directly using off-the-shelf deep networks(Liu et al., 2021). Episode reward, which gives feedback once an episode, is also regarded as sparse reward. Consequently, it is difficult for an agent to find a beneficial path of action because it receives few useful reward signals to guide its actions. Besides, deep networks are usually more difficult to train in reinforcement learning tasks due to their large number of parameters.

To tackle these issues, we proposed a single-step reward function and a lightweight network. The former, which provides timely feedback after each decision made by an agent, can efficiently guide the agent to the optimal viewpoints. Furthermore, we design a lightweight network for extracting image features. The lightweight network with few parameters tends to converge more steadily than deep networks in DOD tasks.

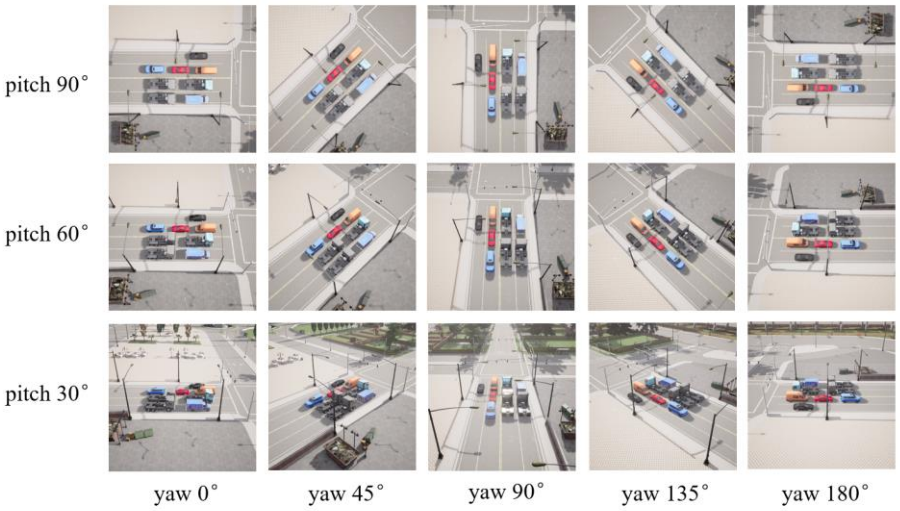


Figure 2. Images captured from different viewpoints

2. Dynamic Object Detection Dataset

A dynamic object detection dataset (DODD) based on UE4 is proposed to demonstrate the effectiveness of the proposed method. DODD includes five object categories: vans, cars, trailers, box trucks, and SUVs. Sample objects for the five categories are shown in Figure 3.

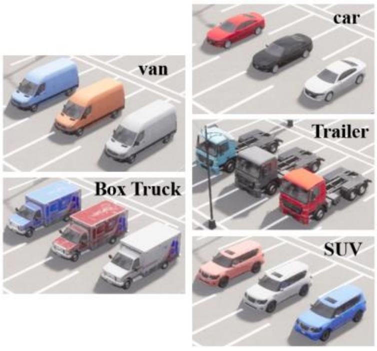


Figure 3 Sample objects

The image acquisition example is shown in Figure 4. Images from 15 viewpoints were captured for each scene, including data from 5 different yaw angles and 3 different pitch angles. The average yaw angle of objects within a scene is set as the initial yaw angle, at which yaw is equal to 0 degrees. The viewpoint directly above the objects has a pitch angle of 90 degrees. At each viewpoint, images are taken from five different distances to simulate different scales. 30m, 40m, 50m, 60m, and 70m from objects correspond to scale 1 to 5 respectively, which means object size is largest in scale 1.

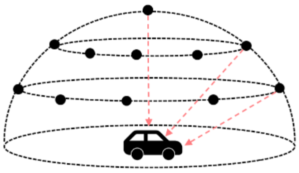


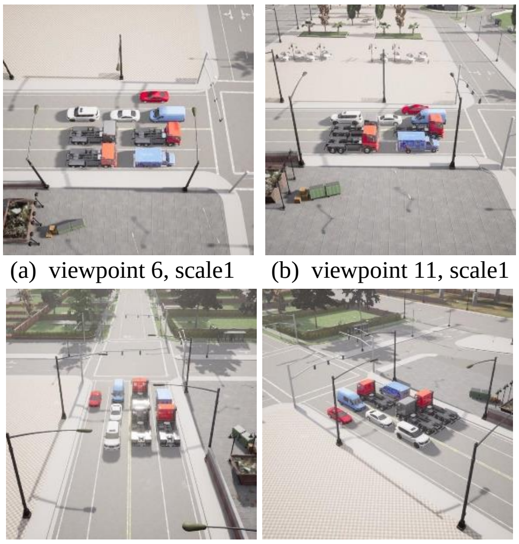
Figure 4 Image acquisition example.

As a result, there are 75 images captured within a single scene. Details of yaw angles and pitch angles are listed in Table 1.

yaw pitch	0°	45°	90°	135°	180°
90°	1	2	3	4	5
60°	6	7	8	9	10
30°	11	12	13	14	15

Table 1 Serial number of each viewpoint.

The length and width of the images in DODD are both 640 pixels. Image samples of DODD are shown in Figure 5.



(a) viewpoint 6, scale1 (b) viewpoint 11, scale1

(c) viewpoint 13, scale1 (d) viewpoint 14, scale1

Figure 5. Image samples.

Statistically, DODD consists of 30 scenes, of which 24 are divided into training scenes and 6 are divided into testing scenes. Consequently, there are 1800 training images and 450 testing images in total. The number of objects per category in the training dataset and test dataset is shown in Table 2.

	car	SUV	trailer	van	box truck
train set	3375	2925	1875	2100	2175
test set	600	675	525	450	1200

Table 2 The number of objects per category in the train set and test set of DODD.

3. Methodology

In this section, we describe the Dynamic Object Detection method we have proposed. The overall architecture of the proposed method is first presented, along with its component modules. Then, the problem definition is presented for the dynamic object detection task, and the training method is explained. The next-view selector and the single-step reward are then presented in turn.

3.1 Dynamic Object Detection Framework

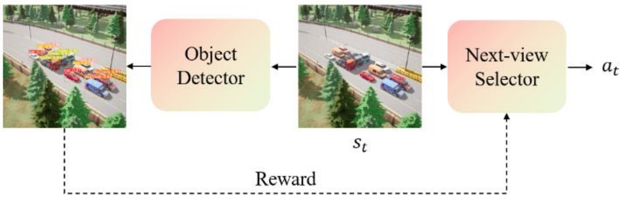


Figure 6 The framework of dynamic object detection.

DOD framework is presented in Figure 6. DOD consists of an object detector and a next-view selector. The object detector provides detection results towards the current image, and the next-view selector adjusts the viewpoint and scale based on the detection result to achieve a better detection score in the next image. We use an off-the-shelf SOTA object detector and design an effective next-view selector to obtain a better detection score.

3.2 Problem Formulation

We consider the dynamic object detection task as a standard model-free reinforcement learning problem. The next-view selector is defined as an agent whose goal is to obtain a better detection score for the object detector by adjusting the viewpoint and scale over several discrete times. We denote S, A, R, P and γ as state space, action space, reward function, transition function and discount factor, respectively.

The state space S includes all the states that the agent may encounter. The action space A consists of seven actions, which are yaw increase, yaw decrease, pitch increase, pitch decrease, scale increase, scale decrease, and stop. Every episode, consisting of several exploration steps of the agent, is ended once exploration steps exceed the episode length or when the agent selects the stop action. We set episode length to 5 in our experiments.

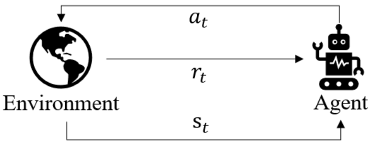


Figure 7 The interactive process between environment and agent in reinforcement learning

The interactive process of reinforcement learning is shown in Figure 7. In discrete time step t, the agent receives the state s_t from the environment and then the action a_t is drawn from the agent’s policy function distribution: $a_t \sim \pi(\cdot\,|s_t)$. After the agent takes a_t, the state s_{t+1} at next time step are obtained from the transition function $P(s_{t+1}|s_t, a_t)$. Meanwhile, $r(s_t, a_t)$ is obtained from the environment after the agent takes a_t at the state s_t. γ reflects the impact of current action a_t on future decisions, which means the current action a_t should not only benefit the next step but also contribute to the overall goal.

As for training, the two-stage training framework is used for the dynamic object detection task. The first stage aims to obtain a well-trained SOTA object detector, which is trained through supervised learning paradigms. The subsequent stage is to train an effective next-view selector, which is trained through reinforcement learning paradigms.

3.3 The Next-view Selector

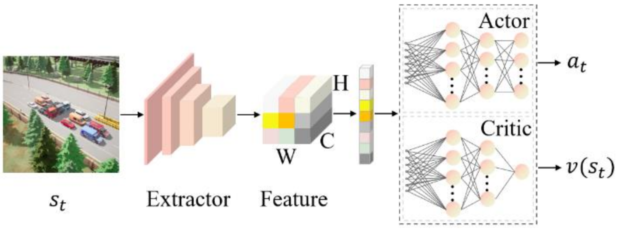


Figure 8 The next-view selector

As shown in Figure 8, the next-view selector consists of a feature extractor network and a policy network. The feature extractor network extracts the image features and the policy network receives the image features to perform decision making. The feature extractor network and the policy network are presented first, and then the PPO algorithm is introduced for the training.

3.3.1 The feature extractor: In deep learning tasks, deep neural networks are typically used as feature extractors. VRL3(Wang et al., 2022) achieved better performance in reinforcement-learning tasks only using 5-layer network. Therefore, we designed a lighter convolutional network as a feature extractor and experimentally demonstrated that the lighter convolutional network works perfectly for dynamic object detection.

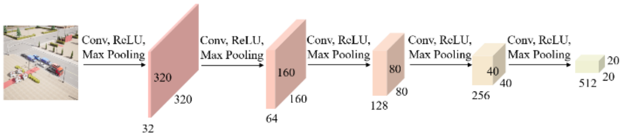


Figure 9 The framework of the lighter convolutional network

The lighter convolutional network is shown in Figure 9, and it consists of five neural network layers, each of which is composed of a 2D convolutional layer, a ReLU layer and a maximum pooling layer. The number of output channels of the five 2D convolutional layers is categorized as 32, 64, 128, 256, and 512. After the average pooling operation is performed on the feature extractor network outputs, they are fed into the policy network.

3.3.2 The policy network: The policy network consists of an actor module and a critic module, where the actor learns the action policy for a given state and the critic evaluates the performance of the given state.

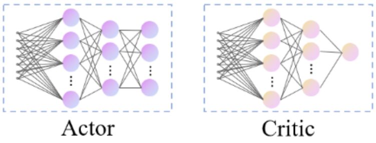


Figure 10 The framework of the policy network. The actor module on the left and the critic module on the right.

As shown in Figure 10, both the actor module and the critic module consist of three neural network layers, corresponding to the input layer, hidden layer and output layer respectively. Each of the layers is composed of a fully connected layer and a Tanh layer. In the actor module, the fully connected layer widths of the input layer, the hidden layer and the output layer are 512, 1024, and 7, respectively. The counterparts in the critic module are 512, 1024, and 1.

3.3.3 Proximal Policy Optimization (PPO) Algorithm: The next-view selector is trained using the PPO algorithm, which is a policy-based reinforcement learning algorithm. The core idea of the PPO algorithm is to limit the gap between the new policy and the old policy when optimizing the policy parameters to ensure the stability of the learning process. Its optimization objective is written as:

$$L^{clip}(\theta) = E_{(s,a)\sim\pi^{\theta_1}}[\min(r_t(\theta)A^{\pi^{\theta_1}}, clip(r_t(\theta),1-\varepsilon,1+\varepsilon)A^{\pi^{\theta_1}})] \quad (1)$$

$$r_t(\theta) = \frac{\pi^{\theta}(a_t \mid s_t)}{\pi^{\theta 1}(a_t \mid s_t)} \quad (2)$$

where θ is the training parameter for the next-view selector, $r_t(\theta)$ represents the update magnitude. The larger $r_t(\theta)$ is, the greater the probability that the current policy π^{θ} will take action a_t in state s_t, and the larger the update relative to the old policy $\pi^{\theta 1}$. $A^{\pi^{\theta 1}}$ is the advantage function under the old policy $\pi^{\theta 1}$ and ε is truncation parameter, which is used to prevent the gap between $\pi^{\theta 1}$ and π^{θ} from becoming too huge.

The key of the PPO algorithm lies in its objective function PPO-Clip, which improves the stability of learning by cropping the ratio between the new policy and the old one so that the new strategy will not deviate too far from the old one. In addition, the PPO algorithm has shown excellent performance and stability in practice and is an effective strategy optimization method.

3.4 The Single-step Reward Function

In reinforcement learning, the sparse reward problem is an important challenge. This refers to the fact that the agent obtains only a few reward signals when exploring an environment, causing learning to become exceptionally difficult. In such environments, the agent may need to go through a large number of exploratory behaviors to get some occasional rewards. This leads to prevent the agent from effectively learning how to accomplish the task.

To mitigate the impact of sparse rewards on the dynamic object detection task, we design a single-step reward function, which provides timely feedback based on the detection results of the current image. Once the agent makes a decision, the precision, recall, and detection confidence are combined to calculate the current reward, which is capable of guiding the agent to adjust the viewpoints to those that are effective for object detection.

$$F_t = P * R * C \quad (3)$$

$$C = \sum_i TP_i \quad (4)$$

where F_t is detection score as time step *t*, *P* is detection precision, *R* is detection recall, and *C* is mean confidence of all true positives in detection results. In order to evaluate the gain of the action on object detection at time step *t*, we set the reward at time step *t* as:

$$r_t = F_t - F_{t-1} \quad (5)$$

Once an episode is ended, a termination reward is obtained, which is as:

$$r_T = 1 - \frac{F_0}{F_T} \quad (6)$$

where *T* means the total steps within an episode.

4. Experiments

In this section, a large number of experiments are conducted to demonstrate the effectiveness of the proposed method. A computer with Intel Core i9-12900K CPU, 64 GB RAM, and NVIDIA GeForce RTX 4090 is used for both training and evaluation.

4.1 Experiments for the Object Detector

Firstly, a SOTA object detector named YOLOv8 is selected as the object detector in the DOD framework, and is trained on DODD.

4.1.1 Hyperparameters for the Object Detector: We give a detailed description of the training hyperparameters for the object detector YOLOv8 here. Firstly, we set the image size to 640x640 pixels, set the batch size to 16, and set training epochs to 40. The Adam algorithm is used as the optimizer, whose momentum and weight decay are set to 0.937 and 0.0005, respectively. Besides, the learning rate is set to 1e-3 at the beginning of training and is reduced to 1e-5 using the cosine decay rate. Lastly, YOLOv8's built-in data augmentation operations are used during training, including horizontal flipping, mosaic enhancement, color translation, and scale translation.

4.1.2 Results for the Object Detector: Firstly, DODD is used to train and evaluate the object detector YOLOv8.

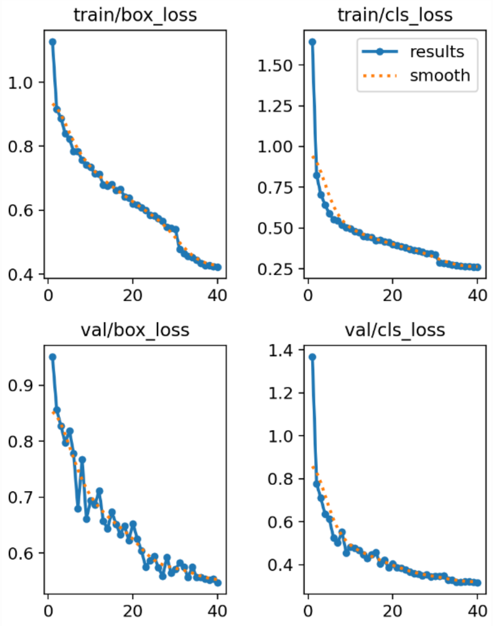


Figure 11 Regression loss and classification loss.

The first and second rows of Figure 11 show the regression loss values and classification loss values for the training and test data, respectively. As can be seen from Figure 11, the object detector YOLOv8 has converged after 40 epochs of training.

For evaluation protocol, Average Precisions(APs) at different *IoUs* (AP at *IoU* =0.50 : 0.05 : 0.95) are used to evaluate the effectiveness of the object detector YOLOv8. As a result, we obtain an AP of 89.1% for the test dataset in DODD.

4.2 Experiments for the Next-View Selector

In this section, the next-view selector in the DOD framework is evaluated based on the feedback of the object detector YOLOv8.

4.2.1 Hyperparameters for the Next-View Selector: We give a detailed description of the training hyperparameters for the next-view selector here. Firstly, the Proximal Policy Optimization (PPO) Algorithm is used to update the parameters of the next-view selector during training. γ, λ, and ε of PPO are set to 0.99, 0.95, and 0.2, respectively. Besides, the learning rate is set to 1e-5 during training. The batch size and minibatch size are set to 1024 and 32, respectively, which means that the next-view selector is trained with a minibatch size of 32 images once the agent explores 1024 steps. Lastly, the total exploration steps of the agent are set to 30k.

4.2.2 Results for the Next-View Selector: Here, we use the proposed DODD dataset to train and evaluate the Next-View Selector.

For the evaluation protocol, the results of all steps within each episode are adopted to evaluate the proposed method. Every image in the test dataset is used as the starting viewpoint. APs are used to measure the performance variations during the viewpoint adjustment in episodes. In order to validate the effectiveness of the proposed method, we compare it to the passive approach and random approach.

Passive Approach(PA) means using the result of the first image in every episode as the episode final result.

Random Approach(RA) means randomly selecting five actions in the action space, executing sequentially and using the result of the last image as the episode final result.

Method	Step					
	0	1	2	3	4	5
PA	89.1	-				
RA	89.1	89.0	89.2	89.0	89.1	89.1
Ours	89.1	92.4	94.4	95.5	95.9	96.0

Table 3 Comparison of Average Precisions(APs) for every step within episodes using different methods.

From Table 3, it can be seen that our method can find a suitable viewpoint for object detection by sequentially adjusting the viewpoint, and our method achieves the best performance compared with PA and ARA. With adjustments according to the proposed method, APs is improved from 89.1% to 96.0% when using the object detector YOLOv8.

	car	SUV	trailer	van	box truck
Step 0	0.88	0.898	0.889	0.886	0.899
Step 5	0.963	0.963	0.937	0.967	0.968

Table 4 Average Precisions(APs) for each category

Average Precisions(APs) , which correspond to step 0 and step 5 with our method, for each category are shown in Table 4. It can be seen that after using the proposed DOD method, the APs of each category are significantly improved.

Lastly, we give an example episode in the testing stage. As shown in Figure 12, the agent chooses the scale increase action two times, chooses the pitch increase action two times, and chooses the yaw increase action one time. As a result, detection score F is increased gradually from 0.7057 to 0.9487.

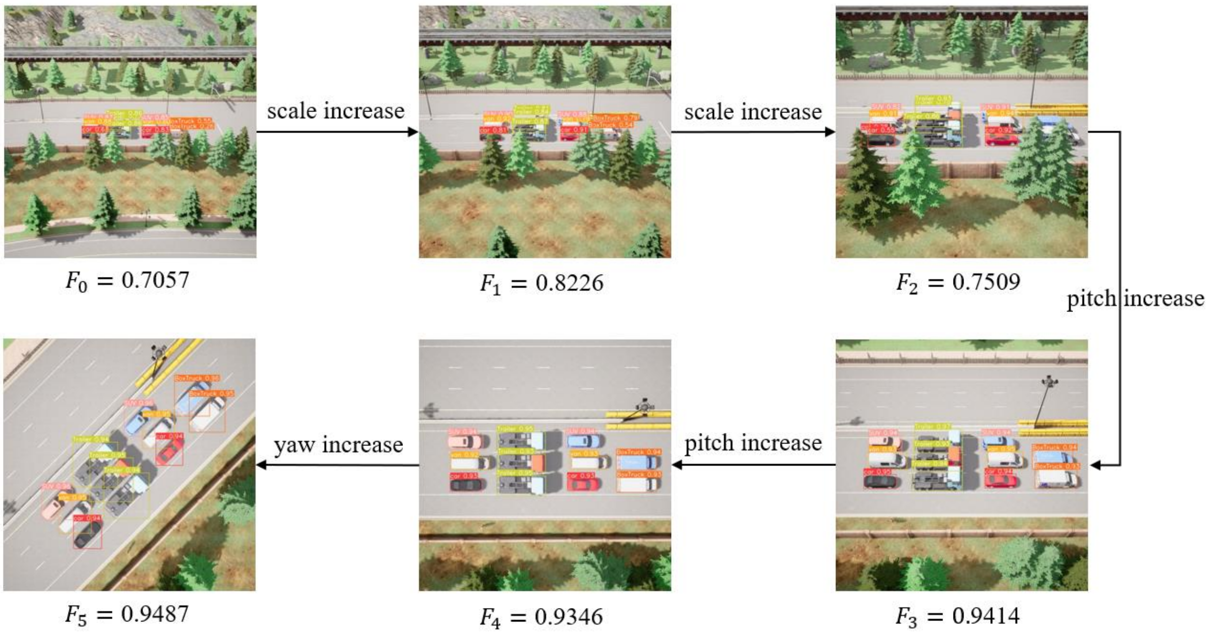


Figure 12 Visualization results of viewpoint adjusting sequence. F of each image is calculated according to Eq.(5)

5. Conclusion

This paper proposed a dynamic object detection method, which adjusts the camera to appropriate viewpoints for the object detection task. Combining precision, recall, and detection confidence, the proposed single-reward function provides a reward signal every time step and effectively alleviates training instability due to sparse rewards. Existing deep networks are over-parameterized, making it difficult for model convergence. Therefore, a lightweight network with few parameters is proposed to improve the speed of convergence efficiently. To demonstrate the effectiveness of the proposed methodology, a DODD dataset containing 1800 training images and 450 testing images is created based on UE4. Finally, the dynamic object detection method is trained using the PPO algorithm. Quantitatively, APs of test images are improved from 89.1% to 96.0% after the viewpoint adjustment by the proposed method when the object detector is YOLOv8.

References

Wang, C., Luo, X., Ross, K., Li, D., 2022. Vrl3: A data-driven framework for visual deep reinforcement learning. Advances in Neural Information Processing Systems, 35, 32974-32988.

Byun, S., Shin, I. K., Moon, J., Kang, J., Choi, S. I., 2021. Road Traffic Monitoring from UAV Images Using Deep Learning Networks. Remote Sensing, 13(20), 4027.

Abdelfattah, R., Wang, X., & Wang, S., 2020. Ttpla: An aerial-image dataset for detection and segmentation of transmission towers and power lines. In Proceedings of the Asian Conference on Computer Vision.

Božić-Štulić, D., Marušić, Ž., Gotovac, S., 2019. Deep learning approach in aerial imagery for supporting land search and rescue missions. International Journal of Computer Vision, 127(9), 1256-1278.

Xu, N., Huo, C., Zhang, X., Cao, Y., Meng, G., Pan, C., 2021. Dynamic camera configuration learning for high-confidence active object detection. Neurocomputing, 466, 113-127.

Ding, W., Majcherczyk, N., Deshpande, M., Qi, X., Zhao, D., Madhivanan, R., Sen, A., 2023. Learning to view: Decision transformers for active object detection. In 2023 IEEE International Conference on Robotics and Automation, 7140-7146.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, 779-788.

Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, 39(6), 1137-1149.

Han, X., Liu, H., Sun, F., Zhang, X., 2019. Active object detection with multistep action prediction using deep q-network. IEEE Transactions on Industrial Informatics, 15(6), 3723-3731.

Liu, S., Tian, G., Zhang, Y., Zhang, M., Liu, S., 2021. Active object detection based on a novel deep Q-learning network and long-term learning strategy for the service robot. IEEE Transactions on Industrial Electronics, 69(6), 5984-5993.