國立成功大學資訊工程學研究所

碩士論文

使用深度強化學習提升物件偵測性能

Deep Reinforcement Learning for Enhancing Object Detection Performance

研究生：郭育丞　　Student: Yu-Cheng Guo

指導老師：賀保羅　　Advisor: Paul Horton

National Cheng Kung University,

Tainan, Taiwan, R.O.C.

Thesis for Master of Science Degree

July, 2023

中華民國一一二年七月

使用深度強化學習提升物件偵測性能

郭育丞* 賀保羅†

國立成功大學資訊工程學研究所

摘要

近年來，深度學習技術在影像處理和分析領域中取得了顯著的進展，尤其在物件偵測方面，取得了顯著的成果。然而，經過訓練的物件偵測神經網路的性能在很大程度上取決於影像品質，如何提高偵測的精度仍然是一個具有挑戰性的問題。影像品質可能受到多種因素的影響，例如圖像模糊、噪音、低對比度、光照不足或過度曝光等。這些問題可能導致深度學習模型難以準確地識別和定位物件。為了克服這些挑戰，本研究提出了一種方法，使用深度強化學習 DDPG 以及 YOLOv7 來實現增強物件偵測的效果。在這種方法中，我們使用前處理技術對影像進行飽和度、亮度、對比度和銳利度處理，以提高影像品質。這些前處理步驟由 DDPG 算法負責執行，以進一步增強影像。接著，我們將經過處理的影像作為 YOLOv7 的輸入，以達到更好的物件偵測效果。我們的實驗結果表明，相較於僅使用單獨的 YOLOv7，這種新的物件偵測方法取得更好的偵測精度。

關鍵詞: 深度強化學習、物件偵測

*學生
†指導教授

Deep Reinforcement Learning for Enhancing Object Detection Performance

Yu-Cheng Guo*   Paul Horton†

Institute of Computer Science and Information Engineering,
National Cheng Kung University

Abstract

In recent years, deep learning techniques have made significant progress in the field of image processing and analysis, especially in the area of object detection. However, the performance of trained object detection neural networks depends largely on the image quality, and it is still a challenging problem to improve the detection accuracy. Image quality can be affected by a variety of factors, such as blurred images, noise, low contrast, insufficient lighting, or overexposure. These problems may make it difficult for deep learning models to accurately identify and localize objects. To overcome these challenges, this study proposes a method to achieve enhanced object detection using deep reinforcement learning DDPG as well as YOLOv7. In this approach, we use preprocessing techniques for image saturation, brightness, contrast, and sharpness to improve image quality. These preprocessing steps are performed by the DDPG algorithm to further enhance the image. Then, we use the pre-processed images as input to YOLOv7 to achieve better object detection. Our experimental results show that this new object detection method achieves better detection accuracy than using only YOLOv7 alone.

Keywords: Deep reinforcement learning, Object detection

*Student
†Advisor

誌謝

在此，我深感榮幸並由衷地對所有在我的學術旅程中給予我幫助的人表示感謝。我要首先感謝我的指導教授 賀保羅教授。他不僅願意接受我轉到他的實驗室，更在他的專業領域之外，毫無保留地引導我完成這篇論文。他對我慷慨的承擔和指導讓我受益良多，而我將永遠珍視這份特殊的學術關係。

我同樣要向我的實驗室同學：益宗、昱伶、中柏、筱萱和怡靜表達深深的感謝。在最後的學期裡，當我面臨挑戰與困境時，他們都伸出援手，提供了我莫大的支援與鼓勵。他們的熱忱、關心與友誼成為我寶貴的支持力量。

最後，我要向我的家人致上最深的謝意。在我攻讀研究所期間，他們給予我金錢與精神上的支持。他們的慈愛與鼓勵為我鋪平了前行的道路，並為我在學術之路上繼續前進提供了堅實的後盾。

這篇論文的完成並非只有我一人的功勞，每一位在我學術旅程中陪伴我的人，都是這份成就不可或缺的一部分。我再次對你們表達最深的感激之情，這是我心中永恆的感謝。

CONTENTS

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

In recent years, deep learning techniques have made remarkable advancements in various fields, particularly in the realm of image processing and analysis. Object detection, in particular, has gained significant attention due to its powerful recognition and localization capabilities, playing a critical role in numerous application scenarios. However, most deep learning models for object detection are trained on high-quality images, and their performance is heavily dependent on image quality. When images are affected by factors such as blurriness, noise, low contrast, inadequate lighting, or overexposure, the detection performance of these models may significantly degrade. Therefore, it becomes an important and challenging problem to enhance object detection accuracy under such non-ideal image conditions.

1.2 Motivation

Despite some existing research efforts to improve object detection performance under adverse image conditions by enhancing model structures or adjusting optimization strategies, these methods often require extensive parameter tuning and cannot guarantee satisfactory results in all situations. On the other hand, there has been limited exploration of using image preprocessing to improve image quality and, consequently, enhance object detection accuracy. Additionally, determining suitable preprocessing techniques and their corresponding parameters remains an unresolved challenge. Therefore, we believe that investigating the integration of deep learning and preprocessing techniques to improve object detection accuracy under adverse image conditions is a worthwhile research topic. To address this issue, we propose the use of Deep Deterministic Policy Gradient (DDPG), a deep reinforcement learning algorithm, to automatically determine the optimal image preprocessing strategy. The goal is to effectively enhance image quality and improve the accuracy of object detection.

1.3 Objective

The main contribution of this research is the first integration of deep reinforcement learning (DDPG) with the object detection model YOLOv7, applied in the image preprocessing stage to enhance object detection accuracy under adverse image conditions. Our approach considers image quality factors such as saturation, brightness, contrast, and sharpness, and employs deep reinforcement learning to automatically select the optimal preprocessing strategy for achieving the best image enhancement results. Through our experiments, we have demonstrated a significant improvement in object detection accuracy compared to using YOLOv7 alone. Our method not only enhances object detection performance but also provides a new direction for effectively handling image quality issues in deep learning.

Chapter 2

Materials and Related Work

2.1 Object Detection

Object detection is an important topic in computer vision that aims to identify the location and classify specific objects in an image. Here is a review of some significant research in this field:

2.1.1 Traditional methods

Before deep learning became mainstream, some traditional methods for object detection have been widely studied and applied. For example, Histogram of Oriented Gradients (HOG) [4] and Deformable Part-based Model (DPM) [5] are two of the important methods.

The HOG algorithm performs object detection by analyzing the local shape information of an image. It first calculates the gradient direction and intensity of each pixel in the image and converts this information into a histogram representation. By counting the gradient direction in the local area, the HOG algorithm can capture the texture and shape features of the object. Combined with the support vector machine classifier, the extracted HOG features can be classified to achieve object identification and localization.

The DPM algorithm treats an object as consisting of several deformable parts and learns the shape and appearance model of each part. It achieves more accurate detection and localization of objects by applying HOG features to each part of the object and establishing a constraint relationship between the parts. Similarly, DPM can be combined with a support vector machine classifier to achieve classification and identification of the extracted features.

2.1.2 Deep Learning methods

With the rapid development of deep learning, a series of neural network-based object detection methods have been proposed. Girshick et al. introduced R-CNN [7], a method that uses selective search to generate candidate regions and then performs feature extraction and classification using convolutional neural networks. Building upon R-CNN, Girshick and Ren et al. further improved the efficiency and accuracy of object detection with Fast R-CNN [6] and Faster R-CNN [13], respectively.

Another significant research achievement is the YOLO [12] (You Only Look Once) series, which was introduced by Redmon et al. YOLO treats object detection as a regression problem and completes object identification and localization in a single inference pass. This approach not only enhances efficiency but also demonstrates excellent performance on various datasets. Multiple versions of YOLO (such as YOLOv2, YOLOv3, YOLOv4, etc.) have further improved the performance of the original model.

You Only Look Once v7 (YOLOv7) [15]: Compared with the previous state of the art YOLO series, YOLOv7 reduces the number of parameters by about 40% and the amount of computation by about 50% and is mainly divided into two aspects: model architecture optimization and training process optimization. For model architecture optimization, the authors propose extended and scaling methods to efficiently utilize parameters and computation; and for training process optimization, YOLOv7 uses re-parameterized and dynamic label assignment strategies, which can assign labels to different output layers more efficiently.

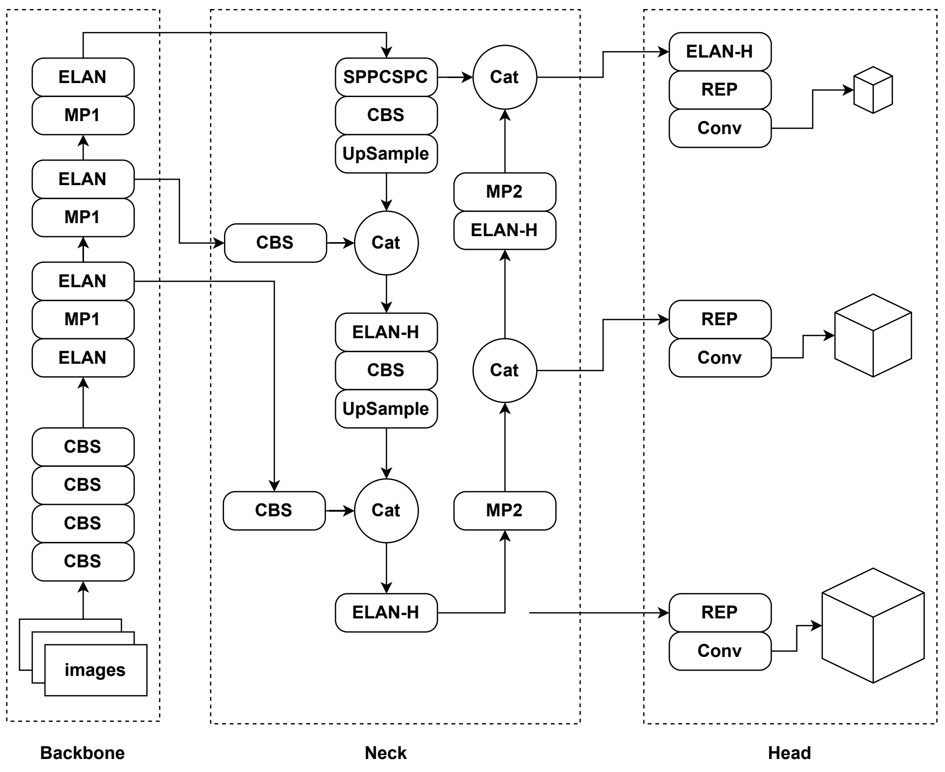


Figure 2.1: YOLOv7 architecture

In summary, research in the field of object detection has shifted from traditional methods to deep learning-based approaches, leading to significant advancements. However, despite the good performance of these methods in many scenarios, challenges still exist in handling complex backgrounds, small objects, and highly overlapping objects. Therefore, object detection remains an active research area that requires further study and development.

2.2 Reinforcement Learning

Reinforcement learning is a method in machine learning that enables models to learn and improve their behavior through interaction with an environment. In the framework of reinforcement learning, a agent takes actions to influence the environment and adjusts its behavior based on feedback from the environment, often referred to as rewards. The goal is to learn a policy that maximizes the accumulated rewards over time. Reinforcement learning has been successfully applied in various fields, including games, autonomous driving, robotics, natural language processing, and more.

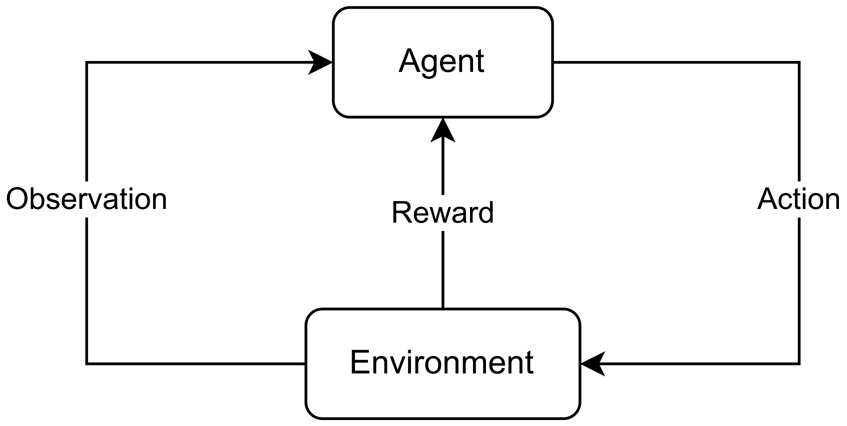


Figure 2.2: Reinforcement learning architecture

Deep Deterministic Policy Gradient (DDPG) [9]: A reinforcement learning algorithm widely used for problems in continuous action spaces. DDPG adopts an actor-critic architecture, which provides separate learning mechanisms for the actor and the critic, allowing for mutual improvement of the policy and value function. In DDPG, the actor is a policy function that aims to find a policy that maximizes the expected future rewards starting from the current state. The actor selects an action based on the current state, and this action influences the next state of the environment. The critic, on the other hand, is a value function that evaluates the goodness or badness of executing a certain action given a particular state. The critic's goal is to learn to predict the impact of the actor's policy on future rewards. The actor and the critic collaborate during the learning process. The actor improves its policy based on the feedback from the critic, while the critic improves its value function prediction based on the actor's policy. This combination enables DDPG to perform well in problems with continuous action spaces.

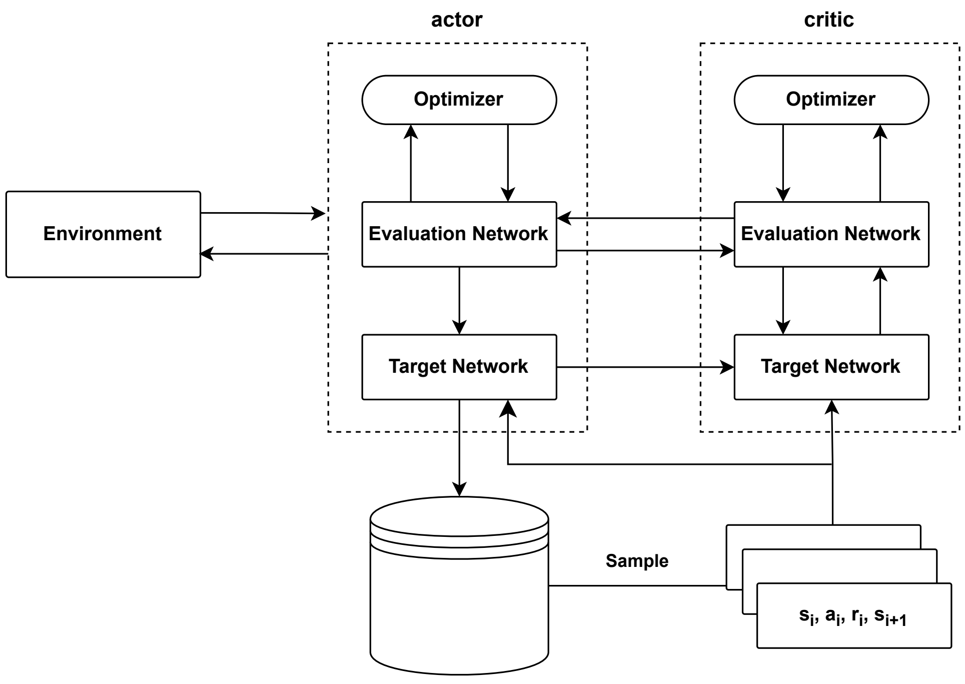


Figure 2.3: DDPG architecture

2.3 Related Work

[3], [2] used DRL for object detection by treating the object detection problem as an MDP and using a hierarchical approach for object detection. In their approach, the Agent is responsible for finding the region of interest in the image and then finding smaller regions from the previously selected regions, thus forming the hierarchy. The authors consider several different actions to improve the accuracy of bounding boxes around the target. They use image feature vectors and action histories to represent states, and IOU transformations as a bonus.

[8] proposed an object detection method called Tree-RL, which utilizes two types of actions: panning (8 actions) and zooming (5 actions). The state is defined as the combination of the feature vector of the current window, the feature vector of the whole image, and the historical actions. Feature vectors are extracted from the ImageNet-trained VGG-16 model and rewarded with

IOU transformations. tree-RL performs object detection by top-down tree search, starting from the whole image and selecting the best action in each window to generate two new windows.

[11] proposed a preprocessing algorithm called ObjectRL. The main idea is to adapt the preprocessing steps applied to the image so that the pre-trained object detector works more efficiently.The novelty of the ObjectRL algorithm is that it recognizes that an image that looks good to the human eye may not be suitable as an input to the pre-trained object detector. In other words, the subjective quality of an image may not be relevant to its object detection performance. Therefore, ObjectRL tries to select the most suitable image for the object detection task by learning appropriate preprocessing strategies. This work is similar to our work.

[16] proposed a reinforcement learning method based on the Deep Q-Network (DQN), significantly reducing redundant search steps through a novel reward function design. During the entire localization process, the detector is considered as an agent that adjusts the size of the bounding box for object localization and employs a terminal action classifier to determine when to cease searching. Furthermore, the researchers applied multi-task learning to enhance the agent's performance. Experimental results showed that this approach demonstrated exceptional performance on the PASCAL Visual Object Classes Challenge (VOC) 2007 dataset.

[1] introduced a reinforcement learning agent called ”Bounding-box Automated Refinement” (BAR) that aims to enhance object detection and recognition accuracy using deep learning techniques. The BAR agent was specifically designed for industrial applications to improve production efficiency. By learning from human demonstrations, BAR autonomously corrects inaccurate annotations, thereby reducing the time and effort required for manual dataset labeling. The study compared two training methods: offline deep reinforcement learning (DRL) and online training using a contextual bandit (CB) approach. The results revealed that both methods successfully im-

proved annotation accuracy and reduced the need for manual intervention, regardless of the initial labeling approach. Overall, the research presented the development of BAR as a reinforcement learning agent that effectively improves the quality of existing bounding boxes, demonstrating its valuable applications in real-world scenarios.

[14] presents an adaptive framework aiming to reduce the cost of using high-resolution images in computer vision tasks. The method utilizes Convolutional Neural Networks (CNNs) for image recognition and object detection, selectively employing high-resolution images only when fine-grained details are necessary. By training an intelligent agent, the framework first processes low-resolution images to capture the global image context and then selectively requests specific high-resolution image patches for accurate detection. Experimental results on the xView dataset demonstrate that compared to detectors exclusively using high-resolution images, the proposed method achieves nearly the same accuracy while using high-resolution images only about 30% of the time, resulting in a roughly 50% increase in runtime performance.

Chapter 3

Problem Definition and Method Selection

Before introducing the method we propose, it is first necessary to define the problem we are aiming to solve and explain why we have chosen to use reinforcement learning as our primary approach.

3.1 Problem Definition

Object detection is a core problem in the field of computer vision, aiming to identify and locate specific objects in given images. From autonomous vehicles to medical image analysis, the applications of object detection have encompassed many aspects of our lives. However, despite significant advancements in this field in recent years, improving the accuracy and efficiency of object detection remains a major challenge.

In real-world application scenarios, the quality and characteristics of images can be influenced by various factors, including lighting conditions, camera angles, environmental variables, and more. These factors can have a significant impact on the features of an image, thereby affecting the effectiveness of object detection. Therefore, in order to improve the performance of object detection, we need to explore methods to automatically adjust the parameters of an image, such as saturation, brightness, contrast, and sharpness, to enhance the image quality and the clarity of its features. We expect that by adjusting the images in this manner, and subsequently applying an object detection model, we can achieve better detection results.

3.2 Method Selection

Finding a method that can automatically adjust image parameters, such as saturation, brightness, contrast, and sharpness, may pose challenges to traditional machine learning methods and supervised learning within deep learning. These methods usually rely on a large amount of labeled data, which may be difficult to obtain or not fully applicable in real-world situations.

To address the challenges mentioned above, we have chosen to use Reinforcement Learning, specifically the Deep Deterministic Policy Gradient (DDPG), as our primary solution strategy. Compared to traditional supervised learning methods, Reinforcement Learning does not rely on a large amount of labeled data. Instead, it learns and adjusts its policy by interacting with the environment. This learning approach allows the model to better adapt to environmental changes and make effective decisions in information-sparse scenarios.

An essential characteristic of Reinforcement Learning is its ability for exploration and exploitation. This capability allows the model not only to rely on the known best strategies during the learning process but also to explore new possibilities to find better solutions. This is particularly crucial for our problem, as the image parameters most suited to improving object detection accuracy might vary under different environmental conditions.

The reasons why we chose to use Deep Deterministic Policy Gradient (DDPG) and the advantages it brings can be primarily categorized into the following points:

- **Self-learning and Optimization** : DDPG can learn from previous experiences and progressively optimize its decision-making strategy. This allows the model to maintain high efficiency in constantly changing environments.

- **Capability of Handling Highly Continuous Action Spaces** : DDPG has the ability to handle highly continuous action spaces. This is especially important in image processing applications where image parameters (such as saturation, brightness, contrast, and sharpness) need fine-tuning.

Chapter 4

Proposed Method

4.1 System Design

The primary objective of this study is to improve the accuracy of object detection. We propose a method that combines Deep Deterministic Policy Gradient (DDPG) with YOLOv7 to enhance the effectiveness of object detection. This chapter will provide a detailed introduction to the system's design and implementation. Figure 4.1 shows the complete system flow chart.

4.1.1 Preprocessing with DDPG

In the entire system workflow, we first preprocess the image. This preprocessing stage involves using a technique called Deep Deterministic Policy Gradient (DDPG) to automatically adjust the image. Initially, we input the original image into the DDPG model. The DDPG model generates four parameters based on the image information, which are used to adjust the image's saturation, brightness, contrast, and sharpness. These parameters are known as transformation factors. Subsequently, we use these four transformation factors to modify the original image and generate an adjusted image, which is then input into the object detection model, YOLOv7.

During the model training process, we utilize the precision and recall of the image as the indicators derived from the output of the YOLOv7 model. These indicators are used to train the DDPG model, thereby enhancing the capabilities of DDPG. By continuously adjusting the transformation factors, the DDPG model learns how to generate better transformation parameters to optimize the performance of the object detection model.

Based on the four outputs obtained from the DDPG model, we can use numerical methods to adjust the brightness, contrast, saturation, and sharpness of an image. We define α_{br}, α_{co}, α_{sa}, and α_{sh} as the transformation factors for brightness, contrast, saturation, and sharpness generated by DDPG, respectively. For a given pixel intensity (I) at coordinates (x, y), we use the following formula for adjustment:

- **Brightness**: The brightness of an image can be changed by a factor α_{br} as follows:

$$I(x, y) = \text{min}(\alpha_{\text{br}} I(x, y), 255)$$

- **Contrast**: The contrast of an image can be changed by a factor α_{co} as follows: We define $gray()$ to convert an image into a grayscale image.

$$I_{\text{gray}} = \text{gray}(I)$$
$$I(x, y) = \text{min}(\alpha_{\text{co}} I(x, y) + (1 - \alpha_{\text{co}})\text{mean}(I_{\text{gray}}), 255)$$

- **Saturation**: The saturation of an image can be changed by a factor α_{sa} as follows:

$$I(x, y) = \text{min}(\alpha_{\text{sa}} I(x, y) + (1 - \alpha_{\text{sa}}) I_{\text{gray}}, 255)$$

- **Sharpness**: The sharpness of an image can be changed by a factor α_{sh}

as follows: We define $blur()$ as blurring the image

$$I_{\text{blur}} = \text{blur}(I)$$
$$I(x,y) = \min(\alpha_{\text{sh}} I(x,y) + (1-\alpha_{\text{sh}}) I_{\text{blur}}, 255)$$

4.1.2 Object Detective with Yolov7

In the object detection module of the system, we use the pre-trained YOLOv7, an object detection model that can locate and classify objects simultaneously in a single inference. In this module, we take pre-processed images as input for object detection, and YOLOv7 analyzes and processes the images and outputs the results, including the object location and category. These outputs can be used to evaluate the performance of the object detection model, which we measure using two metrics: recall and precision. Recall indicates the ability of the model to correctly detect the actual objects that are present, while precision indicates how many of the objects detected by the model are actually correct.

During the training process, we feed both the original and adjusted images into YOLOv7. Each image will produce corresponding metrics: recall and precision. We then use these metrics to compute the corresponding rewards through the reward function. These rewards are utilized for subsequent learning in DDPG. More detailed explanations will be introduced in the following summary.

4.1.3 System Integration

In the system, we integrate the preprocessing module and the object detection module, which improves the accuracy of object detection. The specific workflow is as follows: Firstly, the original image undergoes image adjustments through the preprocessing module. The preprocessing module utilizes the DDPG model to generate adjustment parameters based on the image information, and applies these parameters to the image, resulting in an adjusted image. The adjustment parameters include transformation factors for saturation, brightness, contrast, and sharpness. Next, the adjusted image is inputted into the object detection module for object detection. The object detection module employs the YOLOv7 network for inference, and performs object localization and classification based on the image features. The output of the model includes the positions and class information of the detected objects.

In summary, in the entire system, we integrate the preprocessing module and the object detection module to enhance the accuracy of object detection. By adjusting the image through the preprocessing module and inputting the adjusted image into the object detection module, we can obtain more accurate object detection results.

4.2 DRL Model

In the following section, We will detail the deep reinforcement learning method we adopt, including the architecture of DDPG and the design principles of state, action, and reward.

4.2.1 Network Archtecture

Figure 4.2 is a network architecture diagram of DDPG, including the feature extraction network Resnet18 and the two-layer fully connected layer.

(i) **Feature Extraction Network - ResNet18**:
To extract important features from the input image, we utilize ResNet18 network. This is a deep residual network characterized by the inclusion of a large number of residual blocks in its architecture, effectively preventing the problem of gradient vanishing during training of deep networks. ResNet18 consists of 18 convolutional layers that process the

input image through specific operations (such as batch normalization, ReLU activation, and residual connections), generating feature maps. These feature maps can be regarded as a high-dimensional representation of the input image, containing information about object shapes, textures, colors, and other details.

(ii) **Two Fully Connected Layers for Learning**:
After feature extraction through ResNet18, the generated feature maps are sent to two fully connected layers for further processing. The function of these fully connected layers is to learn useful features from the feature maps and make decisions. In our scenario, this decision involves adjusting the saturation, brightness, contrast, and sharpness of the image in order to improve the accuracy of object detection.

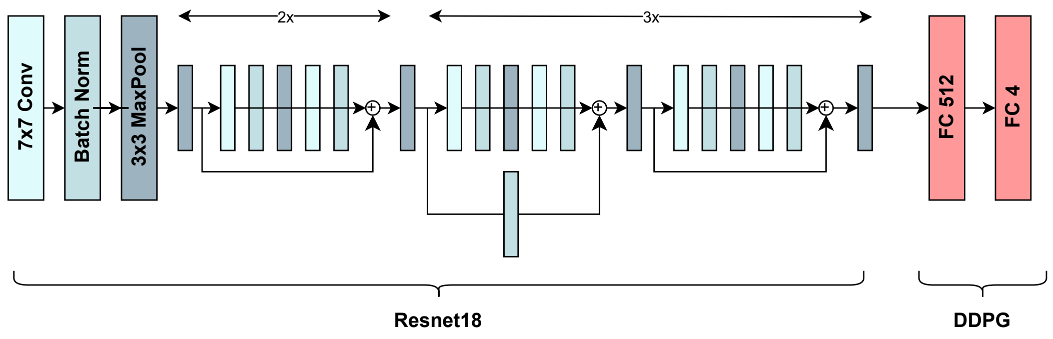


Figure 4.2: DDPG network architecture

Our primary motivation for integrating ResNet18 and fully connected layers is due to ResNet18's capability of extracting deep features from images. These features encapsulate multi-dimensional information about an object, such as its shape, texture, and color. The fully connected layers then learn how to make effective decisions based on the feature maps extracted from ResNet18. This includes adjusting elements like image saturation, brightness, contrast, and sharpness to enhance specific features within the image or make other image adjustments, thereby improving the accuracy and efficiency of object detection.

4.2.2 State

The state of the agent is derived from RGB images from the COCO dataset [10]. Before inputting them into the DDPG model, the images are resized to a dimension of 224x224. This resizing step ensures uniformity in the input dimensions within the system for subsequent processing and analysis.

4.2.3 Action

The actions outputted by DDPG consist of four transformation factors: brightness, saturation, contrast, and sharpness. The values of these factors are constrained within the range of $\alpha \in [0.5, 1.5]$. By restricting the range of the transformation factors to 0.5 to 1.5, we ensure that the enhancement process does not result in excessive image distortion.

4.2.4 Reward

For an image inputted into YOLOv7, we evaluate the score using two metrics: the recall and the precision. By using the following formula, we can calculate the score of the image. We set $\beta = 0.5$.

$$S_i = \beta(\text{precision}_i) + (1 - \beta)(\text{recall}_i)$$

With the formula for calculating the score of an image, we can now present the complete formula for the reward function:

$$r_t = \gamma \tanh(S_r - S_o)$$

In this formula, S_r represents the score of the image after the action transformation, and S_o represents the score of the original image. The difference between S_r and S_o indicates the variation in scores between the two images. A positive difference indicates that the adjusted image improves the detection performance of YOLOv7, while a negative difference suggests a negative impact on the detection performance. To ensure a smoother reward value, we use the tanh function. Additionally, the parameter γ is adjusted to control the curve of the tanh function, and in this case, γ is set to 4.0.

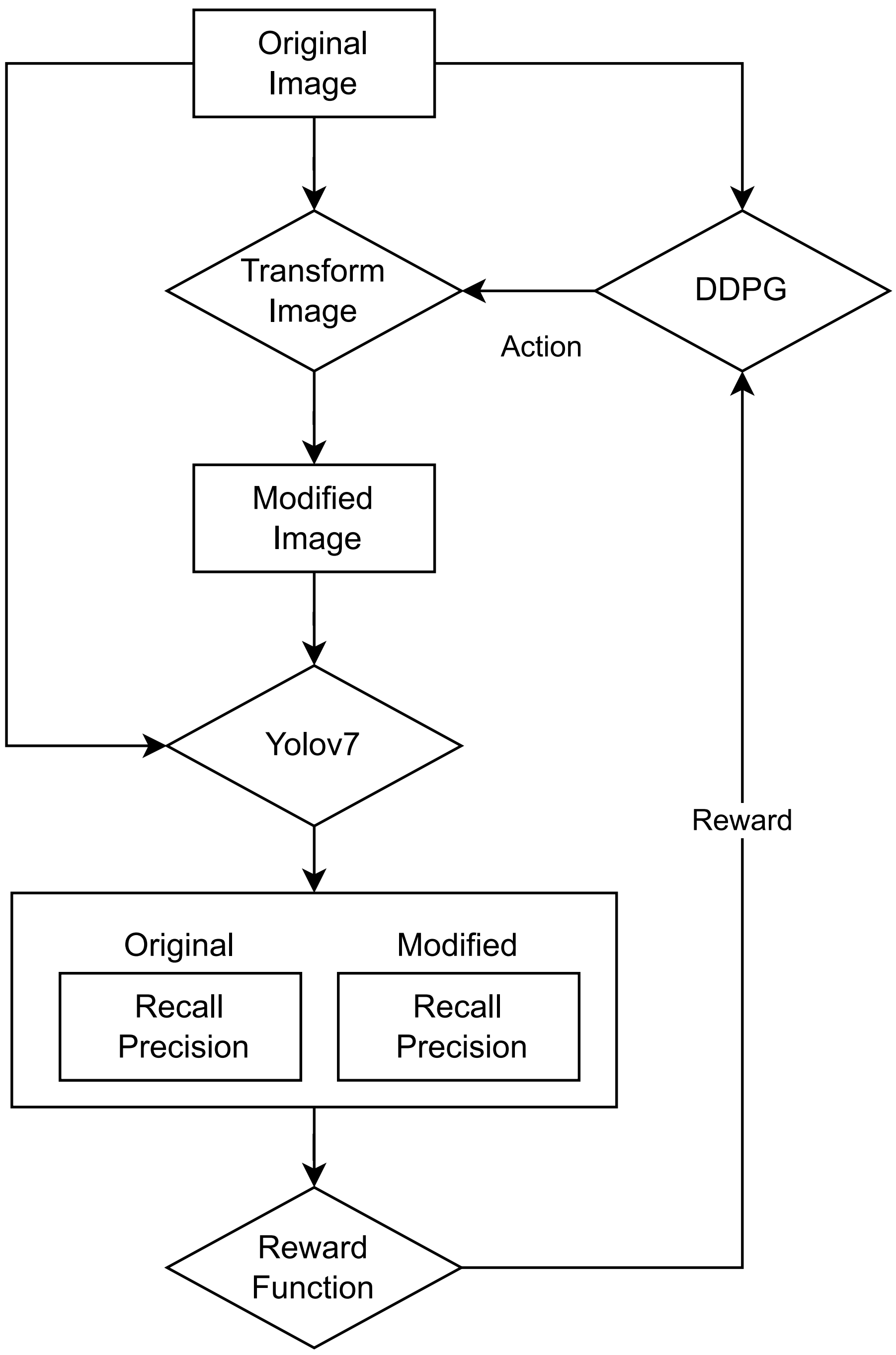


Figure 4.1: System workflow

Chapter 5

Performance Evaluation

In this section, we will introduce the content of the experiment, including the experimental design, experimental environment, and discuss the results of the experiment.

5.1 Experimental Setup and Design

In our experiment, we used the Coco dataset [10] as the training set. Due to the large scale of the Coco dataset and limited experimental resources, we randomly sampled 20,000 data instances for training. A total of 10 epochs were conducted. We built our neural network using PyTorch and trained it using a GeForce RTX 3090. Table 5.1 presents the training parameters for DDPG.

During the training process, we primarily monitor changes in rewards. After training is completed, we proceed with testing. In our testing experiment, we utilize 5000 images, observing the score difference between the adjusted images and the original ones, and recorded the distribution of conversion factors. We will also perform a paired sample t-test to assess the validity of the test. Ultimately, we will compare the performance differences between object detection models preprocessed with DDPG and those without DDPG preprocessing.

Table 5.1: DDPG Parameters

Parameter	**Value**
Episode	10
Replay buffer	4000
batch size	32
Actor network lr	0.003
Critic network lr	0.003
Discount factor	0.99
Neurons per layer	512, 4
Optimizer	Adam
Activation function	ReLU
Loss function	MSE

5.2 Experimental Result

After designing and conducting the experiments, the next step is to analyze and interpret our results. In this section, we will present our experimental results in detail and delve into the underlying reasons for these outcomes.

5.2.1 DDPG Training Reward

As shown in the figure 5.1, after 10 epochs of training, our reward finally converged to 0.05. Our reward function is designed to easily generate so-called sparse rewards. In many images, the reward value is 0. There may be two reasons for this: one is that the objects in the image may have been completely detected, so no matter how the image is adjusted, its precision and recall will not increase, resulting in a reward of 0; another possible reason is that DDPG has not learned how to effectively adjust the image, thus getting the same detection result, and therefore the reward is 0. These two possible factors lead to a convergence of reward value at a lower level. However, although the reward seems to converge only to a relatively small value, it is still greater than 0, implying that DDPG has learned how to adjust the image to improve the effect of object detection.

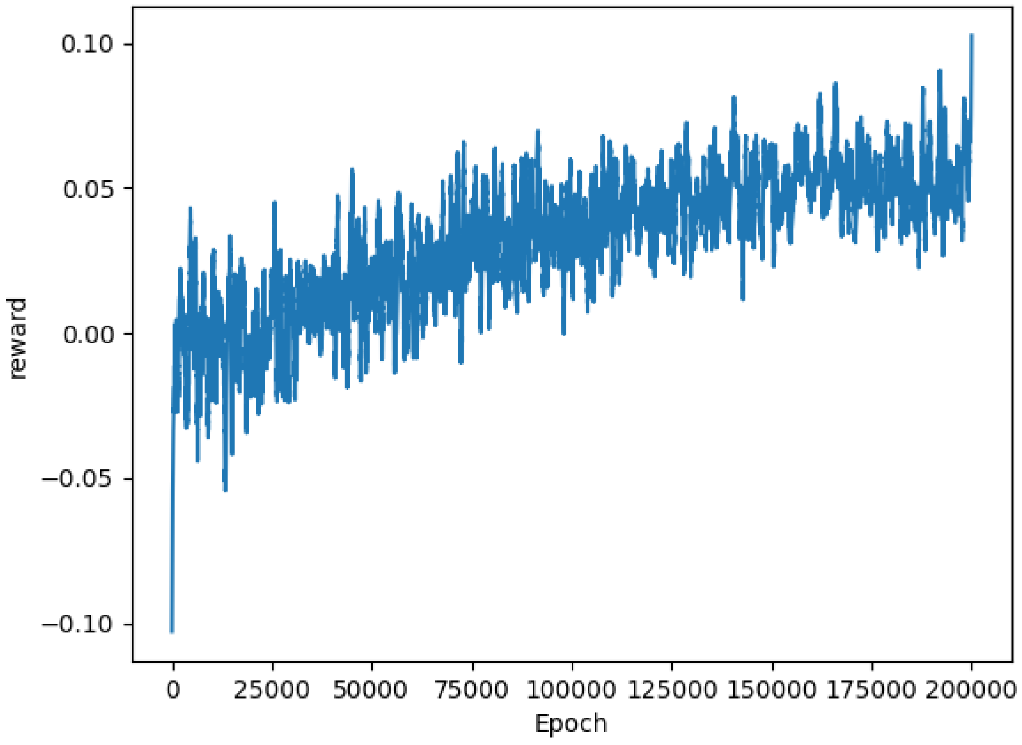


Figure 5.1: Training reward

5.2.2 Test Score Differences with DDPG

Among our 5,000 test images, more than 4000 images had a score difference of zero. This may suggest that in these images, DDPG's preprocessing did not change the result of object detection. For the remaining approximately 900 images, we observed a difference in scores, of which 691 images had a positive score difference, and 203 images had a negative score difference, as shown in Figure 5.2.

A positive score difference implies that the image preprocessed by DDPG is superior to the original image in terms of object detection results, while a negative score difference means the detection effect is worse. However, we can see that among the images with a score difference, the number of images with a positive score difference is significantly more than the images with a negative score difference. This result indicates that using DDPG for preprocessing can improve the effect of object detection in some cases. However, we also noticed that for a portion of images (about 4% of the test images), DDPG preprocessing may lead to a decrease in object detection efficiency. This could indicate that in certain special cases, the DDPG model has not yet found an appropriate preprocessing strategy.

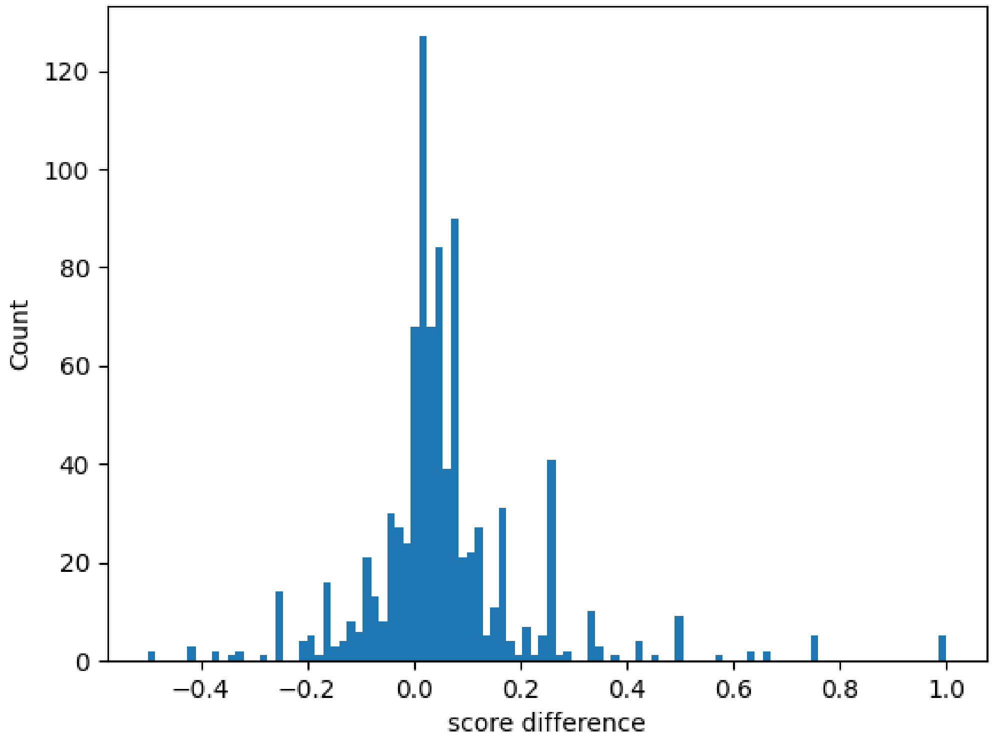


Figure 5.2: Test difference of score

	Mean	**Standard Deviation**
Score Difference	0.057	0.158

Table 5.2: Mean and standard deviation of score difference

In our analysis of the score differences, we also calculated the average and standard deviation of these differences. We found that the average score difference was 0.057, indicating that overall, the object detection scores of images preprocessed by DDPG slightly improved compared to the original images. The standard deviation of the score differences was 0.158, which shows a relatively large range of score differences, suggesting that the impact of DDPG preprocessing on object detection varies among different images.

5.2.3 Test Action Distribution of the Trained DDPG

During the testing process, we recorded various actions taken by DDPG. The results are shown in Figure 5.3.

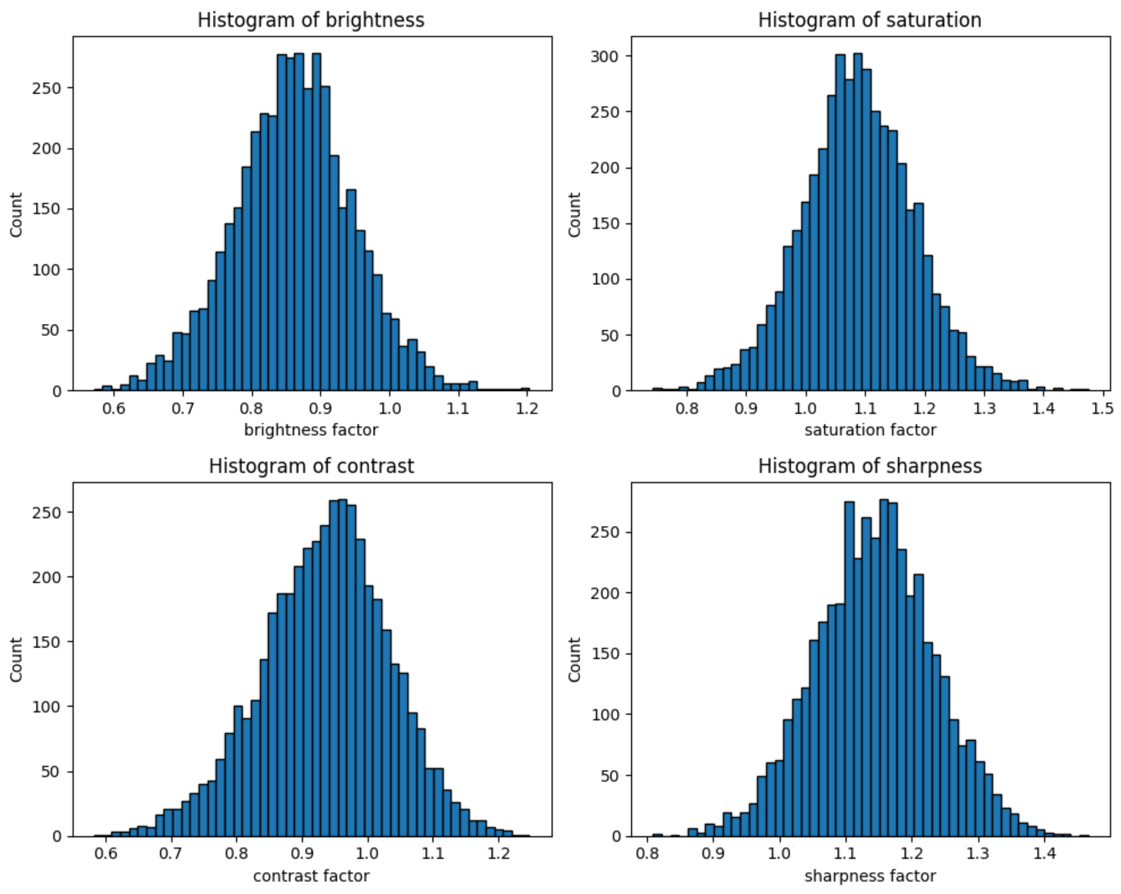


Figure 5.3: Action distribution

Table 5.3: Mean and standard deviation of action distribution

	Mean	**Standard Deviation**
Brightness	0.86	0.09
Saturation	1.09	0.09
Contrast	0.94	0.10
Sharpness	1.14	0.09

The results indicate that DDPG prefers to slightly decrease image brightness (with an average value of 0.86), which may enhance the model's ability to identify image content. DDPG also tends to slightly increase saturation (with an average value of 1.09), emphasizing colors and enhancing the distinction between objects and backgrounds. Meanwhile, DDPG slightly reduces contrast (average value of 0.94), preventing overexposure or underexposure issues and enhancing object features. In addition, DDPG leans towards increasing image sharpness (average value of 1.14), making the image details more distinct. The standard deviation for each transformation factor is close to 0.1, indicating the stability of DDPG's adjustment strategy.

5.2.4 Paired Sample t-Test Results and Implications

In order to quantify these observations and determine statistically whether DDPG preprocessing has had an effect on object detection, we performed a one-tailed paired sample t-test. In our paired sample t-test, we regarded the original score of each image and the score after DDPG preprocessing as a pair of matched samples. We chose a one-tailed test because we expected that DDPG preprocessing would improve object detection.

To correctly interpret the results of the paired sample t-test, we must consider the meanings of the t-statistic and the p-value. In this experiment, the t-statistic was 8.803, which quantifies the degree of difference between the scores of the original images and the scores of the images after DDPG processing. The sign of the t-statistic reflects the direction of this difference - in our case, a positive t-statistic indicates that the average score of the images processed by DDPG is higher than the average score of the original images.

Next, we consider the p-value, which is a measure used statistically to determine whether the results are significant. In this one-tailed test, the p-value is the probability of observing our actual data (or more extreme data) under the null hypothesis. In this experiment, the null hypothesis is ”the score after DDPG preprocessing is lower than the original image score”. We obtained a p-value of 1.86e-18, which is a very small value, much lower than the general significance level threshold (such as 0.05 or 0.01). Therefore, we have sufficient evidence to reject the null hypothesis and conclude that the score of images after DDPG preprocessing is higher than that of the original images.

Table 5.4: One-tailed paired t-test results for image score

	t-statistic	**p-value**
Paired Samples	8.803	1.861e-18

5.2.5 DDPG Preprocessing in Object Detection: A Comparison

To evaluate the impact of DDPG preprocessing on the performance of the object detection model, we separately compared the performance of the object detection models that used DDPG preprocessing and those that did not on the two main performance indicators: precision and recall. As shown in Figure 5.4, the object detection model processed by DDPG is 2.3% higher in precision and 0.9% higher in recall than the model not processed by DDPG.

These results show that DDPG preprocessing can improve the performance of object detection. Specifically, the improvement in precision means that our model's ability to predict correct objects has improved, reducing false positives. The improvement in recall implies that our model can more comprehensively identify objects in the image, reducing the occurrence of false negatives.

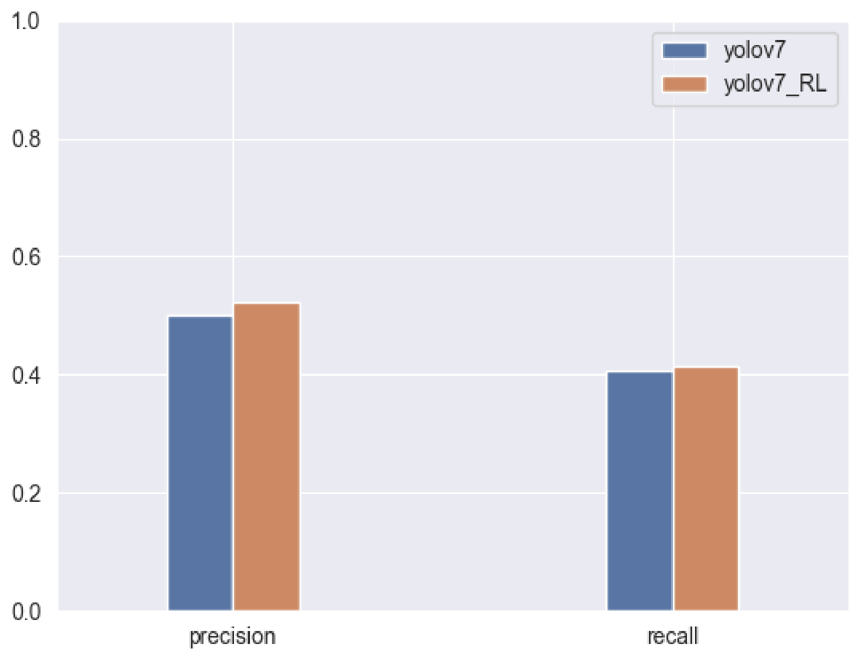
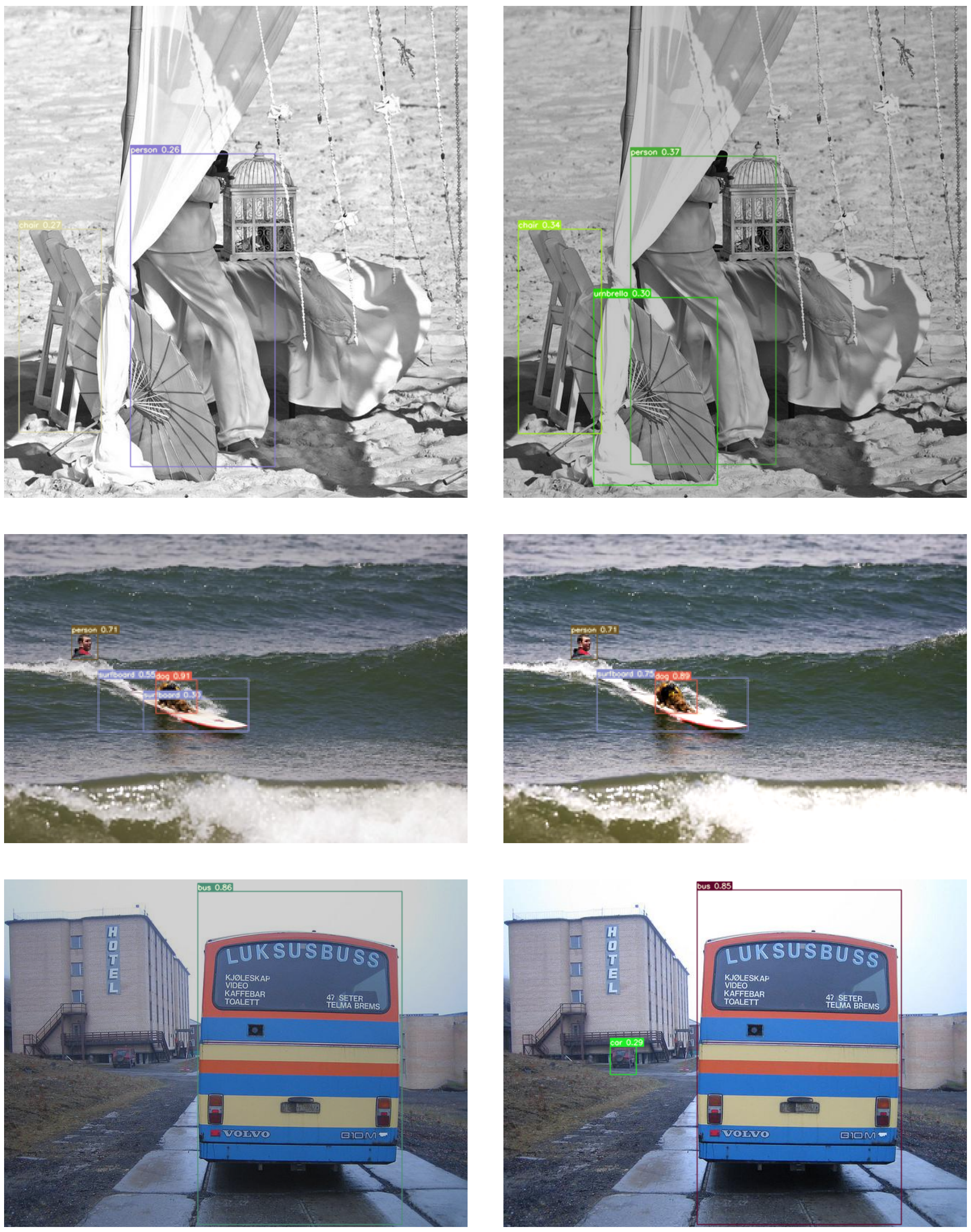


Figure 5.4: Comparison of Object Detection Performance with and without DDPG Preprocessing

Table 5.5: Comparison of yolov7 and yolov7_RL

	Precision	**Recall**
yolov7	0.50	0.406
yolov7_RL	0.523	0.415

5.3 Successful Example



(a) Original image (b) Adjusted image

Figure 5.5: Successful example that demonstrate the difference between the original images (left) and their DDPG-adjusted versions (right). It's noteworthy that objects that were not detected in the original images are successfully identified in the processed images.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this research, we proposed and tested a novel object detection method that combines deep reinforcement learning DDPG with the object detection model YOLOv7. Our results indicate that image preprocessing with DDPG can improve the precision and recall rates of object detection. However, this method might reduce detection effectiveness in some instances. Future studies should focus on further optimization of the DDPG model to cater to various image conditions. Additionally, our experiment was limited by the scale of our dataset, an issue we plan to address in future research by expanding our training and testing sets. Overall, our study provides a fresh perspective on improving the performance of object detection.

6.2 Future work

In our future research, we plan to explore several key directions to improve our approach. Firstly, we intend to compare the speed of object detection with and without the application of DDPG preprocessing. This is a critical issue because while our method can enhance the precision and recall rates of object detection, if it significantly increases processing time, its utility in practical applications might be limited. Therefore, we aim to further optimize our model to ensure that it enhances performance without excessively consuming computational resources.

Secondly, we plan to use data augmentation during the training of the DDPG model. Data augmentation, through various transformations such as rotation, scaling, translation, and others, aims to increase the model's generalization capabilities. We believe this could help the model adapt better to different imaging conditions and potentially further enhance the performance of object detection.

Lastly, we aim to compare our approach with other studies, such as the method proposed in [11]. Through such comparisons, we can gain a deeper understanding of the strengths and weaknesses of our method under different circumstances, and find potential areas of improvement. This will not only help us refine our approach but also contribute to advancing research in the field of object detection.

Bibliography

[1] Morgane Ayle et al. “Bar—a reinforcement learning agent for bounding-box automated refinement”. *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 03. 2020, pp. 2561–2568.

[2] Míriam Bellver Bueno et al. “Hierarchical object detection with deep reinforcement learning”. *Deep Learning for Image Processing Applications* 31.164 (2017), p. 3.

[3] Juan C Caicedo and Svetlana Lazebnik. “Active object localization with deep reinforcement learning”. *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2488–2496.

[4] Navneet Dalal and Bill Triggs. “Histograms of oriented gradients for human detection”. *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*. Vol. 1. Ieee. 2005, pp. 886–893.

[5] Pedro F Felzenszwalb et al. “Object detection with discriminatively trained part-based models”. *IEEE transactions on pattern analysis and machine intelligence* 32.9 (2009), pp. 1627–1645.

[6] Ross Girshick. “Fast r-cnn”. *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[7] Ross Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[8] Zequn Jie et al. “Tree-structured reinforcement learning for sequential object localization”. *Advances in neural information processing systems* 29 (2016).

[9] Timothy P Lillicrap et al. “Continuous control with deep reinforcement learning”. *arXiv preprint arXiv:1509.02971* (2015).

[10] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755.

[11] Siddharth Nayak and Balaraman Ravindran. “Reinforcement learning for improving object detection”. *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer. 2020, pp. 149–161.

[12] Joseph Redmon et al. “You only look once: Unified, real-time object detection”. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[13] Shaoqing Ren et al. “Faster r-cnn: Towards real-time object detection with region proposal networks”. *Advances in neural information processing systems* 28 (2015).

[14] Burak Uzkent, Christopher Yeh, and Stefano Ermon. “Efficient object detection in large images using deep reinforcement learning”. *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2020, pp. 1824–1833.

[15] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 7464–7475.

[16] Yan Wang et al. “Multitask learning for object localization with deep reinforcement learning”. *IEEE Transactions on Cognitive and Developmental Systems* 11.4 (2018), pp. 573–580.