Multi-class Object Detection in Urban Scenes Based on Deep Learning

Yunning Wang [1], Xianglei Liu [1], Runjie Wang [1]

[1] School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture, Beijing 102616, China - 1210081367@qq.com(Y.W); liuxianglei@bucea.edu.cn(X.L); wangrunjie@bucea.edu.cn(R.W)

Keywords: Urban scene; Multi-class object; Target detection; Deep learning; YOLOv5s

Abstract

The rapid development of urbanization presents challenges and requirements for multi-class object detection in urban scenes. Accurately identifying buildings, vehicles, and trees in urban scenes can optimize urban planning, traffic management, monitoring environmental conditions, and promote the development of smart cities. Traditional target detection methods perform poorly in complex urban environments, while deep learning technology achieves accurate target recognition and positioning by automatically extracting high-level semantic features. In this study, we chose to use the YOLOv5s algorithm for multi-class target detection in urban scenes. YOLOv5s is a lightweight deep learning model with small storage space and efficient detection speed. In this paper, the Potsdam area data published by ISPRS is used to make the label data of buildings, vehicles and trees. The YOLOv5s algorithm is used to iteratively train the model. The results show that the mAP value detected by the YOLOv5s model can reach 82.83%. The experimental results show that the algorithm shows higher accuracy than SSD and Faster R-CNN in tree detection. Although it has a slight decline in building and vehicle detection, considering the factors such as detection accuracy, speed, and model size, the YOLOv5s algorithm has a better recognition and detection effect for the detection of multi-class targets in urban scenes.

1. Introduction

In recent years, the acceleration of urbanization has made object detection in urban scenes more important and challenging. Accurately detecting and identifying multiple types of targets in cities, especially buildings, vehicles, and trees, is critical for urban planning, traffic management, environmental monitoring, and smart city development. Accurate detection of buildings can help urban planners formulate scientific and reasonable urban development plans. Accurate detection of vehicles provides strong support for traffic violation monitoring and intelligent transportation system construction. Tree detection in the city can evaluate the ecological environment of the city and formulate corresponding greening plans and protection measures.

Traditional urban scene object detection methods, such as Haar feature and Adaboost algorithm classifier (Krishna et al., 2012), HOG feature and SVM classifier (Wang et al., 2019), bag-of-words model (Tu et al., 2018), often rely on hand-designed features and rules, and the detection speed is slow. Moreover, these methods have poor performance in dealing with complex urban scenes such as illumination changes, occlusion, and scale changes.

The rapid development of deep learning technology provides a powerful tool for multi-class object detection in urban scenes. It can automatically extract high-level semantic features from images or videos to achieve accurate identification and positioning of multiple types of targets in complex scenes. The multi-class object detection algorithm based on deep learning in urban scenes is mainly divided into two types: two-stage algorithm and one-stage algorithm. Two-stage algorithm first extracts candidate regions, and then performs classification and regression operations on these regions to achieve target detection. The well-known two-stage algorithms include R-CNN (Girshick et al., 2014), Fast R-CNN (Girshick et al., 2015) and Faster R-CNN (Ren et al., 2017). One-stage algorithm simplifies the target detection process, no longer needs to generate candidate regions in advance, but directly uses a single convolutional neural network (CNN) for feature extraction, target regression and classification. The main one-stage algorithms are SSD (Liu et al., 2016) and YOLO (Redmon et al., 2016; Redmon et al., 2017; Redmon et al., 2018; Bochkovskiy et al., 2020). Experimental results show that the two-stage target detection algorithm performs well in detection accuracy, but the one-stage algorithm is faster (Jiao et al., 2019).

In target detection, YOLO series algorithm is the most widely used algorithm. When the YOLO algorithm performs target detection, most of the frames and accuracy values detected per second are better than mainstream algorithms such as SSD and Faster R-CNN. The YOLOv5 algorithm in the YOLO series algorithm is only 27 MB under the PyTorch architecture. It is more lightweight and can run on smaller storage space and low-power devices. The YOLOv5s algorithm is the smallest model in the YOLOv5 series of algorithms and is suitable for environments with limited computing resources. Moreover, it is specially designed to provide fast processing speed while maintaining high detection accuracy, which makes it very suitable for applications that require real-time processing. Considering comprehensively, this paper chooses YOLOv5s (Zhao et al., 2023) to detect multiple types of targets in urban scenes.

2. Materials and Methods

2.1 Materials

The data used in this study is a high-resolution urban remote sensing image of 6000 * 6000 pixels in the Potsdam region of Germany published by the International Society for Photogrammetry and Remote Sensing (ISPRS). The Potsdam region of Germany is a typical historical city in Germany, containing a large number of buildings, vehicles and trees. Due to the large image resolution and size in the data set, in order to reduce the amount of calculation and improve the training efficiency, and in order to improve the performance of the model, the data augmentation method is used to effectively expand the number of samples. In this study, the dataset was cropped and divided into a 10 * 10 grid. The original data is shown in Figure 1, and the preprocessed data is displayed in Figure 2.

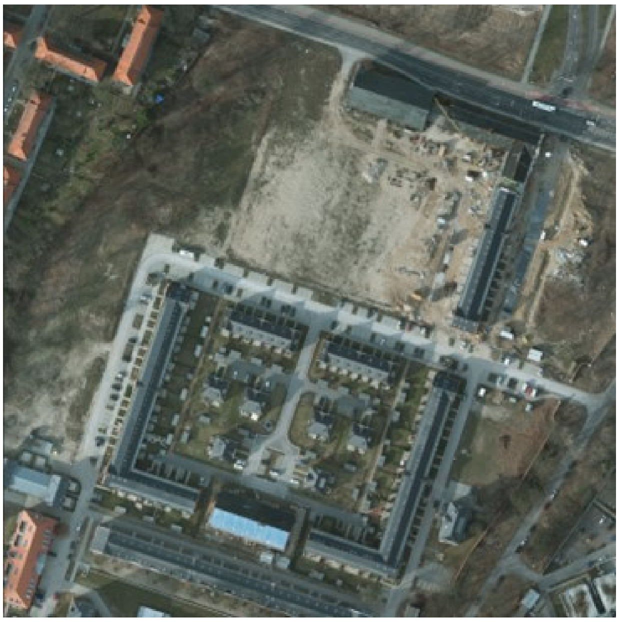

Figure 1. The original data of Potsdam region

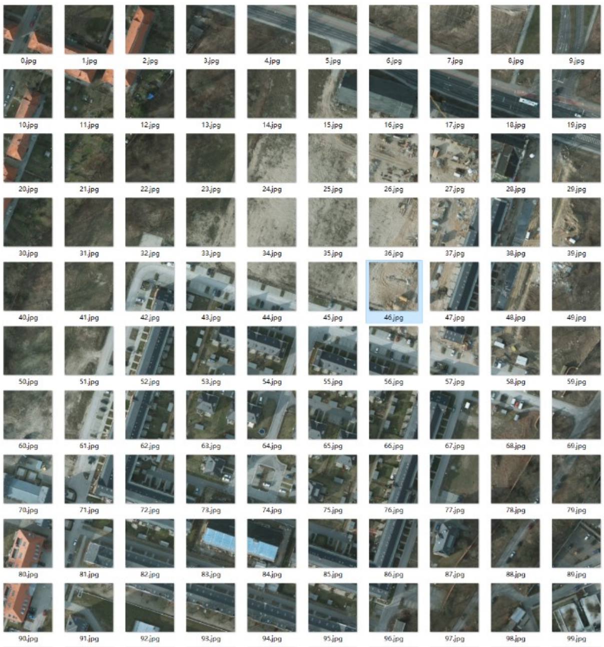

Figure 2. The preprocessed data of Potsdam region

In the process of creating dataset labels, 584 labels were selected as the training set after accounting for incomplete images where buildings, vehicles, and trees were removed, as well as images with missing detection targets. Some of these label images are shown in Figure 3.

The images in the dataset are divided into a training set and a validation set in a 4:1 ratio. In the Dataset folder, the 'Annotations' and 'JPEGImages' folders are used to store labels and image data, respectively, while the 'ImageSets' folder is used to store the text files generated by the algorithm. The structure of the multi-class object detection dataset for urban scenes is shown in Figure 4.

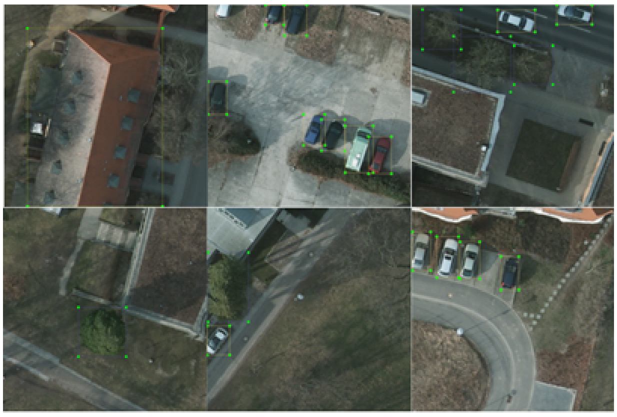

Figure 3. Examples of the training set

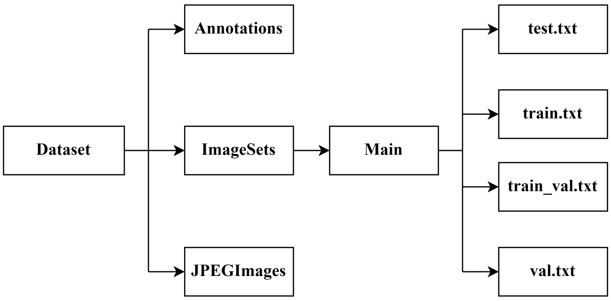

Figure 4. Structure of urban scene object dataset

2.2 YOLOv5s network

YOLOv5 includes four versions: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The four versions of the model structure are basically the same, the main difference lies in the different values of model depth multiplier and model width multiplier. According to the comparison of the prediction results trained by the same data set, the YOLOv5s model has the best speed and performance, while the YOLOv5x model has the best average accuracy (Bochkovskiy et al., 2020; Xing et al., 2021). Moreover, the YOLOv5s model has the simplest structure, the fastest running speed, and the least computational resource consumption. Considering all factors, the version used in this paper is YOLOv5s, and its network structure is shown in Figure 5.

2.2.1 Input: The input side utilizes image preprocessing, so that the input image can be adjusted to the size that the network can input. In the training stage of the network, the optimal anchor box value of different training sets can be calculated by adaptive anchor box computation, so as to optimize the training speed and accuracy of the model. Adaptive anchor box computation helps in refining the model's ability to predict object locations more precisely, thus enhancing overall detection performance. Additionally, the dataset is expanded using the Mosaic data augmentation method, where four images are randomly selected for rotation and scaling. This augmentation method helps the model learn from a more diverse set of scenarios, thereby improving its generalization capabilities and performance on unseen data.

2.2.2 Backbone: The backbone network of the YOLOv5s model is composed of CSPDarknet53, which includes 53 convolutional layers and a SPP (Spatial Pyramid Pooling) module for extracting image features. At the same time, the center point and scale predictions use the CSP structure, and there are two types of stacking: 3-fold and 9-fold, to reduce the amount of calculation. The SPP layer (Yu, 2022) is used to increase the receptive field and enhance the separation of network feature representation from context features.

2.2.3 Neck: The structure of Neck is composed of FPN (Huang, 2021; Hou et al., 2021) and PAN (Lin et al., 2017) structure. It has the characteristics of top-down and bottom-up feature fusion. The feature information output by the Backbone is up-sampled and fused with high-level feature information, and then down-sampled to aggregate shallow feature information, which enriches the image feature information.

2.2.4 Head: The Head part outputs three scale prediction maps at the same time, which are suitable for detecting small, medium, and large targets respectively. GIOU loss (Liu et al., 2021) is used as the loss function for image bounding box regression, and the generated target box is filtered by non-maximal suppression (NMS). Finally, the prediction with the highest confidence is output along with the bounding box coordinates.

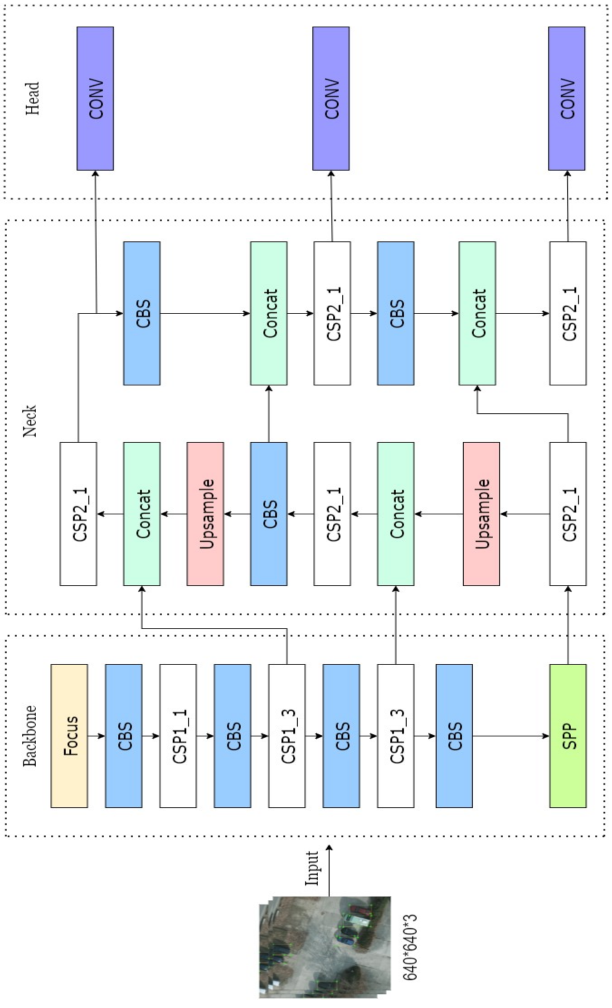

Figure 5. YOLOv5s algorithm architecture

3. Experiment and Results Analysis

3.1 Experimental environment

The experiment uses the Windows 11 operating system and uses the PyTorch deep learning framework to deploy the network model.

The GPU server is used to train the model, and the server configuration is shown in Table 1 and Table 2.

Name	Configuration information
Operating system	Windows 11
Language	Python 3.9
Framework	PyTorch 2.0.1
CUDA	CUDA 12.2
CuDNN	CuDNN 8.9.4

Table 1. Software configuration

Name	Configuration information
CPU	AMD Ryzen 7 PRO 5845 8-Core Processor 3.40GHz
GPU	NVIDIA T1000
RAM	16GB
Storage	1TB

Table 2. Hardware configuration

3.2 Model evaluation

In this paper, the following evaluation indexes are used as the performance evaluation methods of model training: F1 score, mAP (mean Average Precision), FPS (Frames Per Second). FPS is used to evaluate the speed of target detection. That is, the number of images that can be processed per second. The formulas for F1 and mAP are as follows:

$$F1 = 2 \cdot P \frac{R}{P+R} \tag{1}$$

$$mAP = \frac{\sum AP}{N} \tag{2}$$

$$P = \left(\frac{TP}{TP+FP} \right) \tag{3}$$

$$R = \left(\frac{TP}{TP+FN} \right) \tag{4}$$

where, TP (True Positive) represents the number of boxes predicted by the model to be positive and with an IOU greater than a certain threshold with the actual target; FP (False Positive) represents the number of boxes predicted by the model to be positive but less than or equal to a certain threshold with the actual target IOU; FN (False Negative) represents the number of actual targets not detected by the model.

3.3 Results and Analysis

In the training phase of the model, the network parameters are configured as follows: the number of training epochs is 400, the maximum learning rate of the model is set to 0.01, the weight decay coefficient is set to 0.0005, the type of optimizer is SGD, the momentum is set to 0.937, and the batch size is set to 8. After the model training, the mAP curve during the training process is shown in Figure 6, and the training loss is shown in Figure 7. It can be observed that the mAP value increases steadily up to around 350 iterations, with the curve becoming more gradual after 350 iterations, and finally reaches 82.83%.

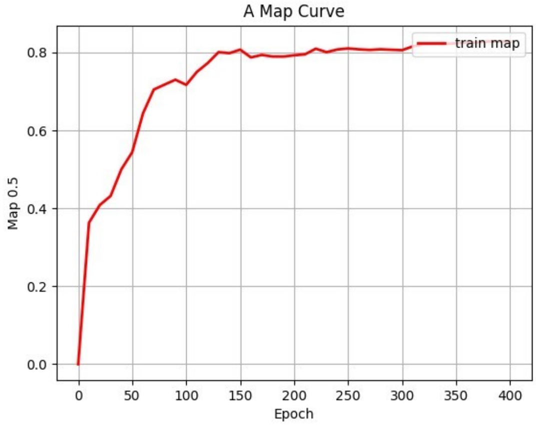

Figure 6. The mAP curve

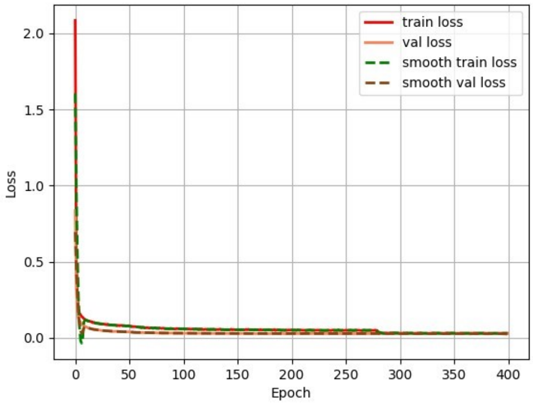

Figure 7. The training loss

When the test set images are input into the previously trained YOLOv5s model for target recognition, satisfactory experimental results are obtained, as shown in Figure 8 and Figure 9. These results show that YOLOv5s performs well in multi-class object detection and exhibits excellent capability in dealing with complex urban scenes. Specifically, the model can accurately identify objects such as buildings, vehicles, and trees that are common in urban environments, providing a reliable basis for scene understanding and analysis. This accuracy is not only evident in single-target categories but also in effectively distinguishing and locating multiple categories simultaneously, providing strong support for the needs of practical application scenarios. This excellent performance not only demonstrates the potential of YOLOv5s as an advanced target detection model, but also provides a robust solution for solving visual tasks in complex real-world environments.

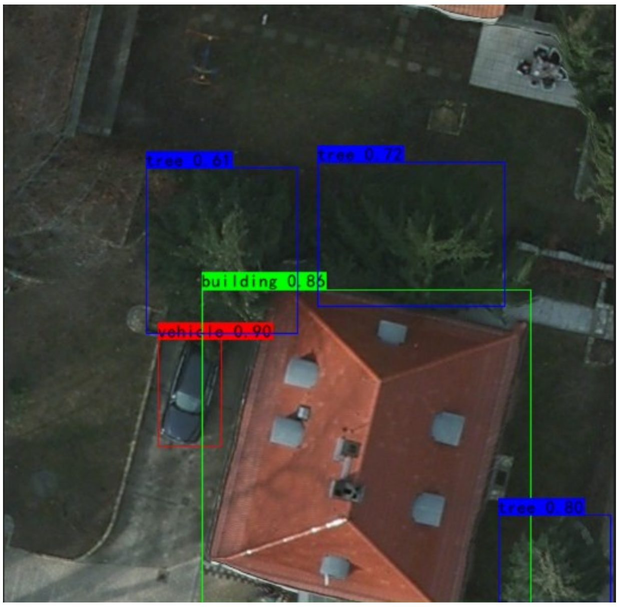

Figure 8. YOLOv5s test result 1

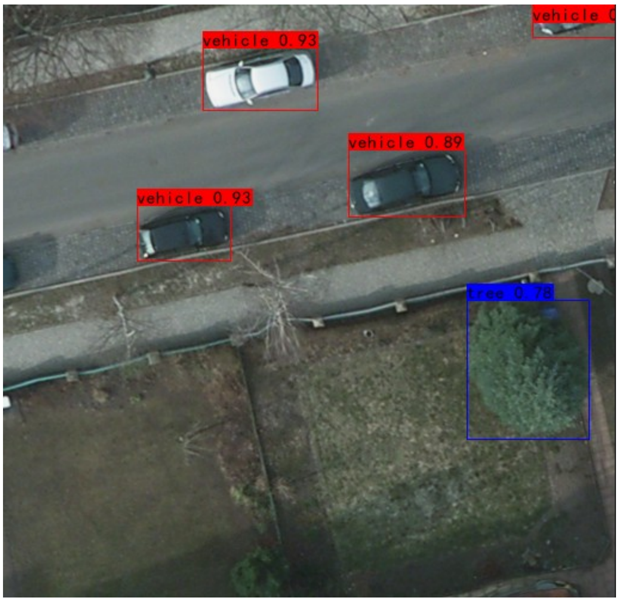

Figure 9. YOLOv5s test result 2

When evaluating the performance of the model selected in this paper, we compare it with the well-known two-stage algorithm Faster R-CNN and the one-stage algorithm SSD, which, like YOLO, is also a single-stage detector. This comparison aims to fully understand the performance of these three models in the target detection task. In the comparative experiment, we pay special attention to the F1 score of the three models, which is an important index for comprehensively evaluating accuracy and recall. Table 3 shows the comparison of F1 scores of the three models, which provides an in-depth understanding of their performance in different scenarios.

In addition, to more comprehensively evaluate the detection effect of the model on different datasets, we further compare their mean Average Precision (mAP). This metric is shown in Table 4, which can reveal the detection accuracy of the model on different types of targets. By comparing mAP, we can gain a clearer understanding of the performance differences among models in various scenarios, providing an important reference for selecting the most suitable model for specific application scenarios.

Beyond accuracy, the number of frames per second (FPS) is also a key indicator for evaluating the practicality of the model, especially for real-time applications. In Table 5, the FPS of the three models is compared to understand their similarities and differences in processing speed.

Category	F1		
	SSD	Faster R-CNN	YOLOv5s
building	0.64	0.57	**0.61**
vehicle	0.83	0.71	**0.80**
tree	0.75	0.50	**0.84**

Table 3. Comparison of three models for target detection F1

Index	SSD	Faster R-CNN	YOLOv5s
mAP	75.78	82.08	**82.83**

Table 4. Comparison of three models for target detection mAP(%)

Index	SSD	Faster R-CNN	YOLOv5s
FPS	24.25	3.74	**44.14**

Table 5. Comparison of three models for target detection FPS

From the F1 results of the YOLOv5s model, compared with SSD and Faster R-CNN, the YOLOv5s model shows higher detection accuracy for trees, but its detection performance for buildings and vehicles are slightly weaker. According to the data, since buildings and vehicles belong to two different scales of targets, they can impact the detection results of the model.

From the results of mAP, the average detection accuracy of the three models, from highest to lowest, is: YOLOv5s, Faster R-CNN, SSD. It can be seen that although the YOLOv5s model has a slightly worse detection result for different scale targets on the F1 evaluation, the result of the model is still well in the overall detection of multi-class targets in urban scenes.

In terms of FPS, the YOLOv5s model has a clear advantage in speed compared to SSD and Faster R-CNN, with an FPS of 44.14. This represents a significant benefit for real-time multi-class object detection in urban scenes.

In general, the YOLOv5s model has certain advantages over SSD and Faster R-CNN in terms of both detection accuracy and speed for multi-class urban scene objects, especially in terms of speed, which meets the requirements of real-time detection in urban scenes.

4. Conclusions

This paper proposes a YOLOv5s algorithm for multi-class target detection in urban scenes. First, buildings, vehicles, and trees, which are common in urban scenes, are selected as the research objects. In this paper, the dataset used is from the Potsdam region of Germany, published by the International Society for Photogrammetry and Remote Sensing. By cutting and preprocessing this urban scene dataset, we obtain a target dataset that is more conducive to labeling and training. Then, the preprocessed dataset is labeled and used for training. Through continuous adjustment of model parameters, we develop a YOLOv5s model that performs well in multi-class object detection tasks in urban scenes. Finally, the trained YOLOv5s model is applied to detect the test dataset. To better understand the results of the model, comparative experiments are conducted with the Faster R-CNN and SSD models.

Based on the experimental results and analysis, we draw the following conclusions: The YOLOv5s algorithm achieves satisfactory results in detecting multiple types of targets in urban scenes. The results show that the mAP value of the YOLOv5s model can reach 82.83%. Although the F1 scores for building and vehicle detection with the YOLOv5s algorithm are slightly lower than those of SSD and Faster R-CNN, the overall accuracy of this algorithm for multi-class target detection in urban scenes is better than that of SSD and Faster R-CNN. Additionally, the detection speed of the YOLOv5s algorithm shows significant advantages over SSD and Faster R-CNN, providing strong support for real-time and intelligent monitoring of urban scene targets. Furthermore, the YOLOv5s algorithm's small size and good robustness enhance its application value and potential in urban scene object detection.

The following further discusses the limitations and optimization directions of YOLOv5s model in urban scene object detection: The dataset used in this paper is a single dataset. In the next step, we can try to add datasets of other urban scene objects for labeling and training to further improve the robustness of the model. In addition, to meet the requirements of real-time performance and intelligence, it is necessary to further optimize the YOLOv5s algorithm to improve the model's detection speed.

References

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M., 2020. Yolov4: Optimal speed and accuracy of object detection. *arxiv preprint arxiv:2004.10934.*

Girshick, R., Donahue, J., Darrell, T., & Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587).

Girshick, R., 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).

Huang, W., Chen, R., Yuan, T., 2021. Improved YOLOv3-SPP UAV target detection model compression scheme. *Computer Engineering and Application*, 57(21):165-173.

Hou, Z., Liu, X., Yu, W., Pu, L., Ma, S., Fan, J., 2021. Improved non-maximum suppression target detection algorithm using GIoU. *Journal of Electronics*, 49 (04): 696-705.

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R., 2019. A survey of deep learning-based object detection. *IEEE access*, 7, 128837-128868.

Krishna, M. G., & Srinivasulu, A., 2012. Face detection system on AdaBoost algorithm using Haar classifiers. *International Journal of Modern Engineering Research*, 2(5), 3556-3560.

Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S., 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117-2125).

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C., 2016. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14* (pp. 21-37). Springer International Publishing.

Liu, Z., Yuan, L., Zhu, M., Ma, S., Chen, L., 2021.YOLOv3 traffic sign detection combining SPP and improved FPN. *Computer Engineering and Applications*, 57 (07):164-170.

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* (pp. 779-788).

Redmon, J., & Farhadi, A.,2017. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263-7271).

Redmon, J., & Farhadi, A., 2018. Yolov3: An incremental improvement. arxiv preprint arxiv:1804.02767.

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.

Tu, J., Sui, H., Feng, W., Sun, K., 2018. Using bag-of-words model to detect damaged areas on the top of buildings. *Journal of Wuhan University (Information Science Edition)*,43 (05):691-696.

Wang, Y., Zhu, X., & Wu, B., 2019. Automatic detection of individual oil palm trees from UAV images using HOG features and an SVM classifier. *International Journal of Remote Sensing*, 40(19), 7356-7370.

Xing, Y. C., Li, D. J., & Ye, F. M., 2021. Remote sensing image target detection based on YOLOv5. *JiangXi Science*, 39(4), 725-732.

Yu, Z., 2022. YOLO V5s-based deep learning approach for concrete cracks detection. In *SHS Web of Conferences* (Vol. 144, p. 03015). EDP Sciences.

Zhao, W., Syafrudin, M. and Fitriyani, N.L., 2023. CRAS-YOLO: A novel multi-category vessel detection and classification model based on YOLOv5s algorithm. *IEEE Access*, 11, pp.11463-11478.