International Journal of Scientific Engineering and Science

Volume 8, Issue 7, pp. 83-88, 2024. *ISSN (Online): 2456-7361*

Pedestrian and Vehicle Detection System Based on Deep Learning

Rongye Wang[1], Yingming Liu[2], Hongjian Chen[3]

[1, 2, 3]School of Information and Electrical Engineering, Shandong Jianzhu University, Jinan, Shandong, China

***Abstract*—*This paper discusses the application of deep learning technology in the field of pedestrian and vehicle detection. Pedestrian and vehicle detection is an important problem in computer vision, which has a wide range of practical applications. We review YOLO algorithms and their applications in pedestrian and vehicle detection. The selection and labeling methods of data sets, model training and optimization strategies, and the application of evaluation indexes are discussed. Finally, we analyze current technology challenges and future directions, including improved detection accuracy, real-time performance, and further improvements in the generalization ability of complex scenarios.*

***Keywords*—*Deep learning, YOLOv4, Object detection*.

I. INTRODUCTION

In recent years, algorithms for pedestrian and vehicle detection based on deep learning have developed rapidly. Compared to traditional object detection algorithms, they offer faster detection speeds and higher accuracy. Many researchers have applied these algorithms in fields such as autonomous driving and intelligent surveillance, achieving promising results. However, in practical applications, challenges arise due to diverse object categories, occlusions between targets, and complex environments.

This system will utilize the YOLOv4 object detection algorithm, enhanced through various improvements to increase detection speed, improve detection of small objects, and enhance robustness in complex environments. This adaptation aims to better suit real-world traffic scenarios. The overall model architecture of YOLOv4 is depicted in the following figure:

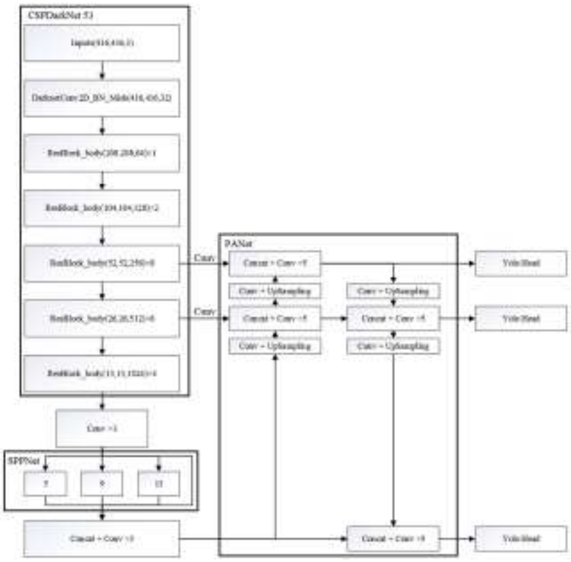

Fig. 1. YOLOv4 model architecture.

II. GENERAL REQUIREMENT

A. System function overview

The system is designed primarily for detecting pedestrians, vehicles, and common objects in road traffic environments. When a camera captures an image, it is passed into the system to identify the category and location of the objects to be detected. Given the complexity of real-world road traffic scenarios, the system requires high detection speed and accuracy, along with robust performance in challenging environments.

B. System functional requirements

In order to optimize the road traffic object detection system, we can adopt several key strategies. First, data enhancement techniques are used to enrich the background of the object to be detected, which helps to prevent the model from over-relying on a specific background during training, thus avoiding overfitting. Secondly, the data set is expanded, especially to include more samples in complex environments, so as to improve the robustness and generalization ability of the model in real scenarios. The backbone feature extraction network is further improved to optimize the detection speed and efficiency. For the detection of small objects, advanced technology is used to improve the detection accuracy and sensitivity of the model to small objects. Finally, the detection results are visualized, and the category information and precise position of the object to be detected in the image are marked, which is helpful to evaluate and adjust the performance of the model. These integrated strategies will significantly improve the accuracy and practicality of the road traffic object detection system.

III. SYSTEM DEVELOPMENT

A. System requirement analysis

System requirement analysis: To provide services for the analysis and decision-making of intelligent driving vehicles and intelligent video surveillance.

System structure design: Based on the analysis of the system functional requirements, the entire system is divided into the following several parts: data enhancement module, backbone feature extraction network, feature enhancement network, YOLO Head, and visualization module.

The system process is shown as follows.

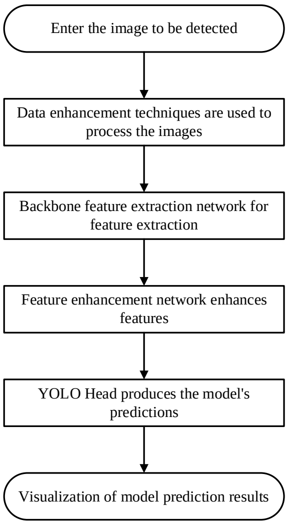
Enter the image to be detected
Data enhancement techniques are used to process the images
Backbone feature extraction network for feature extraction
Feature enhancement network enhances features
YOLO Head produces the model's predictions
Visualization of model prediction results

Fig. 2. YOLOv4 model architecture.

B. *System functional requirements*

Most of the target detection models based on deep learning are written in the Python programming language. The main reason is that Python programming is more concise and convenient compared to C or C++, allowing researchers to focus more on the construction of the model rather than the programming language. Therefore, this model will also be written in Python. In addition, the construction of neural networks is a key point. Currently, the commonly used frameworks for building neural networks include TensorFlow, Keras, PyTorch, etc. Considering the simplicity and ease of use of each framework, Keras is selected as the framework for building the model.

Currently, target detection algorithms are roughly divided into two categories, one is one-stage, and the other is two-stage. Compared to two-step algorithms, one-step algorithms have higher recognition speed and recognition accuracy. Therefore, most target detection algorithms use the idea of one-stage algorithms for target detection. The YOLOv4 target detection algorithm improved by this system is a representative of one-stage algorithms. The core idea of this algorithm is to transform a target detection task into a regression task, that is, input an image into the system, and the position and category information of the object to be detected can be obtained without performing other operations outside the model. This greatly speeds up the detection speed of the model and improves the real-time performance of model detection.

The following figure shows the general process of the one-stage algorithm:

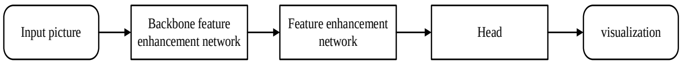
Input picture
Backbone feature enhancement network
Feature enhancement network
Head
visualization

Fig. 3. one-stage flow chart of object detection algorithm.

C. *System overall structure design*

The system was developed using Python programming language and keras framework to build the entire neural network model. The following figure is the improved model based on YOLOv4. In the model, the original network structure was improved to improve the detection speed of the model, strengthen the detection of small objects and enhance the robustness of the model in the face of complex environments. The following is the model architecture after improving YOLOv4.

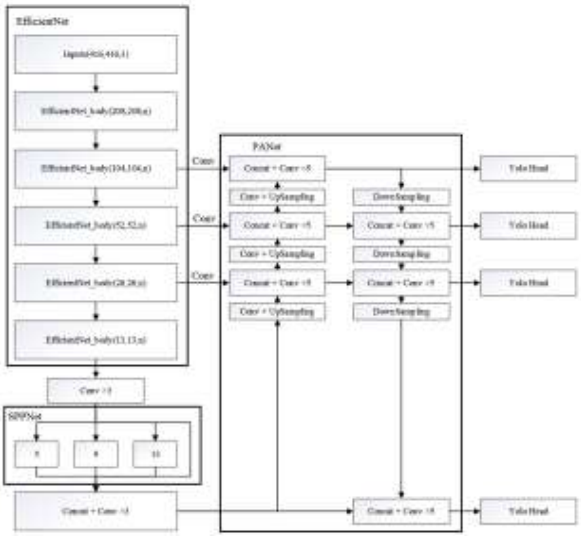

Fig. 4. Improved YOLOv4 model architecture.

IV. THE DESIGN DESCRIPTION OF EACH MODULE IN THE SYSTEM

A. *Backbone feature extraction network*

The target detection algorithms of the YOLO series have been using the Darknet framework as their backbone feature extraction network since YOLOv2. From DarkNet-19 used in YOLOv2 to CSPDarkNet-53 used in YOLOv4, its network structure has been optimized step by step, and the detection effect has also been greatly improved. However, because this network structure is very large, the number of parameters in this network is also very large, which somewhat limits the detection speed of YOLOv4 models. So, is there a network structure with equally superior performance but fewer parameters?

EfficientNet seems to be a good choice. This network minimizes the computational cost and the number of parameters of the model while ensuring the detection accuracy of the model, which is several times less than other models of the same accuracy level. In addition, the EfficientNet series includes multiple versions from EfficientNet-B0 to EfficientNet-B6. The detection accuracy of these version models is getting higher and higher, and the scale of the models is also getting larger. Through continuous experiments to balance the detection speed and detection accuracy, a more suitable model can be selected. Therefore, this article will mainly focus on the detection of pedestrians, vehicles and common objects in the road traffic environment. That is, when a camera captures an image and transmits it to this system, the category information and position information of the object to be detected can be identified. Because it is facing the real road traffic environment, the system is required to have a high detection speed and detection accuracy, and also have good robustness when facing complex environments.

The following figure compares the number of parameters of different YOLOv4 versions after improvement with the original YOLOv4:

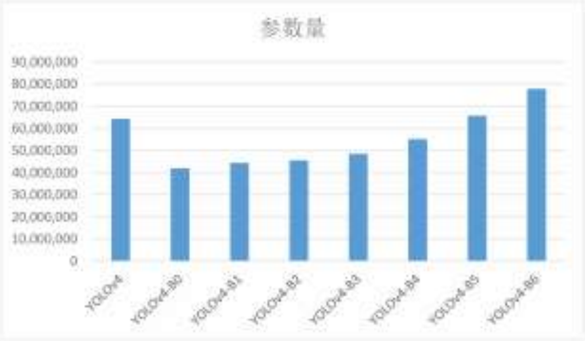

Fig. 5. Parameter quantity comparison chart.

It can be seen from Figure 5 that compared with the original YOLOv4 model, the number of parameters in the improved YOLOv4 model has been greatly reduced, so it can be predicted that the detection speed of the improved model will be improved to a certain extent.

B. Improved detection of small objects

For the target detection task, the detection of small objects has always been a difficult problem. The main reason is that after continuous operations such as convolution and pooling of small objects, the feature map keeps shrinking, resulting in the features of small objects becoming less and less obvious. Usually, after the feature extraction of the convolutional neural network for the image, three effective feature layers will be generated, which are the original image reduced by 32 times, 16 times and 8 times respectively. Since the size of the input image is generally 416×416, the sizes of the three effective feature layers are generally: 13×13, 26×26, and 52×52.

In real road traffic scenarios, distant pedestrians, some roadblocks and markers belong to small objects. Therefore, while the model pays attention to global information, it should also increase the attention to detailed information. In the convolutional neural network, the shallow features contain rich detailed information, while the information of small objects is seriously lost in the deep features. Therefore, when building the network structure, the shallow features need to be utilized, which can improve the model's detection of small objects.

Here, a simple convolutional neural network is established to show the difference between shallow features and deep features. The network architecture is as follows:

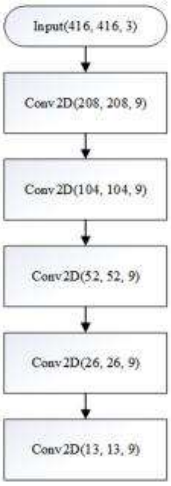

Fig. 6. Convolutional neural network.

Figure 6 is a simple convolutional neural network to show the difference between shallow features and deep features. Here, feature maps of sizes 13×13, 26×26, 52×52, and 104×104 will be displayed respectively. Among them, the 104×104 feature map is a shallower feature compared to the other three. The following picture is the input image:

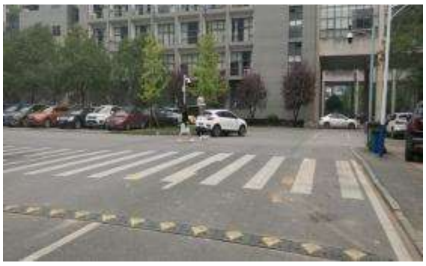

Fig. 7. Input picture.

When the picture of Figure 7 is input into the convolutional neural network, feature maps of sizes 13×13, 26×26, 52×52, and 104×104 can be obtained. Here, visualization techniques are used to visualize these feature maps for easier observation.

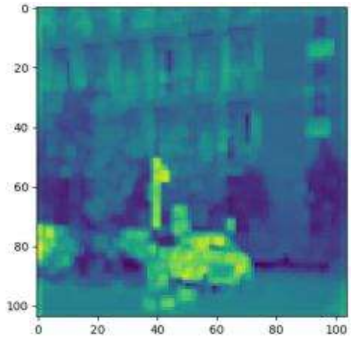

Fig. 8. 104×104 size feature map.

It can be seen from Figure 8 that in the effective feature layer of this layer, the features of pedestrians and vehicles have been abstracted to a certain extent, but many detailed information is still retained. In the first downsampling, the texture and shape information of the input picture can still be seen.

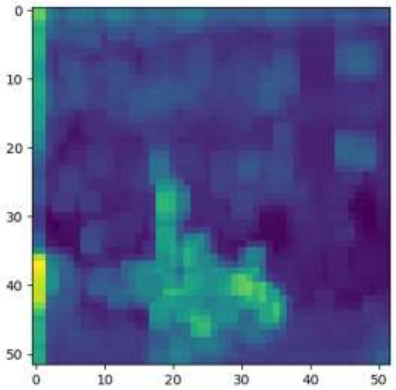

Fig. 9. 52×52 size feature map.

Figure 9 shows the feature map of size 52×52×52. This feature map has undergone two downsamplings, and it can be seen that compared to the feature of size 104×104, the features of pedestrians and vehicles have been further abstracted, and some detailed information has also begun to be lost.

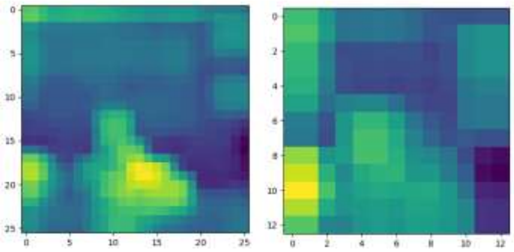

Fig. 10. 26×26 and 13×13 size feature maps.

Figure 10 shows the feature maps of size 26×26 and 13×13. It can be observed that in the feature maps of these two scales, the input image has undergone a large number of convolution and pooling operations, the features of the image have become very abstract, and the detailed information has been completely lost. The features obtained here are more representative.

To enhance the detection of small objects in the YOLOv4 model, it is necessary to make full use of the shallow features and fuse the shallow features with the deep features, so that the model can obtain more detailed information while obtaining abstract information. In this article, four effective feature layers will be extracted from the EfficientNet network, and the sizes of the effective feature layers are: 13×13, 26×26, 52×52, and 104×104. By adding an additional shallow effective feature layer, the detection effect of the model for small objects can be enhanced.

C. Extended data set.

In the real road traffic environment, the target detection task is extremely challenging. The main reasons are as follows: The background in the road traffic environment is complex with many distractions; pedestrians, as non-rigid bodies, have various postures and different clothes; in scenes with dim light such as at night or on rainy days, the features of objects are not obvious, making it difficult to extract features; the size of objects in the road traffic environment is small and they are easily occluded. Therefore, if one wants to train a model that can adapt to the real road traffic environment, it is necessary to prepare a dataset with rich enough background and high enough quality for the model.

This article will use the larger-scale COCO 2017 dataset as the main body and combine the real road traffic environment in Yuelu District, Changsha City, Hunan Province to expand the dataset. The scenes for image collection include: urban streets, traffic light intersections, etc., and the collected weather includes: sunny days, cloudy days, rainy days, etc. The robustness of the model for target detection in complex environments is improved by expanding the dataset.

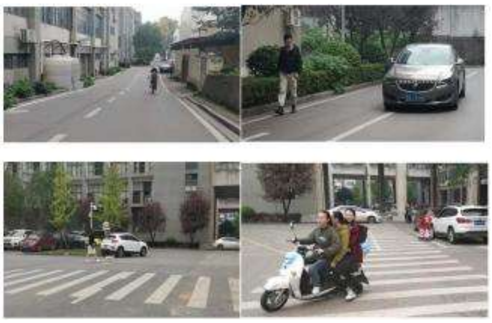

Fig. 11. Capture picture.

D. Using Mosaic data enhancement technology

In computer vision, the commonly used data augmentation operations can be mainly divided into two categories. One is distortion, and the other is image occlusion. Distortion can be divided into illumination distortion and geometric distortion. Illumination distortion refers to achieving the effect of data augmentation by changing the brightness, contrast, saturation and noise of an image. Geometric distortion expands the dataset by randomly scaling, cropping, flipping, rotating, etc. of an image. However, this data augmentation operation will cause the real bounding boxes in the image to be deformed, so they must be re-labeled. Both data augmentation techniques in distortion are pixel-level adjustments and can be restored to the original image through a series of transformations.

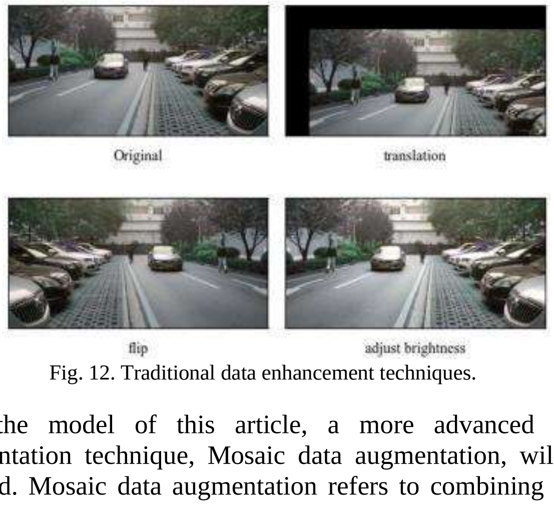

Fig. 12. Traditional data enhancement techniques.

In the model of this article, a more advanced data augmentation technique, Mosaic data augmentation, will be adopted. Mosaic data augmentation refers to combining four training images into one training image according to a certain proportion. The advantage of doing this is that it enables the model to learn how to recognize objects that are smaller than the normal size, and at the same time, it can greatly enrich the background of the detected objects, improving the generalization ability of the model when facing new backgrounds.

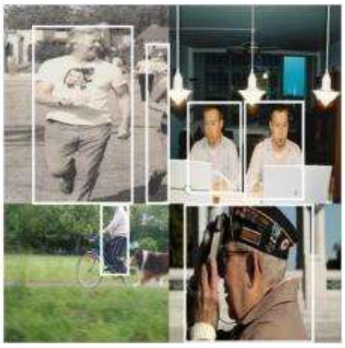

Fig. 13. Mosaic data enhancement.

E. Visualization technique

The visualization technology of this system mainly refers to the system reading videos and presenting the prediction results, which is mainly achieved through OpenCV. When presenting the prediction results, screening is first carried out based on the scores of the prediction boxes to obtain some prediction boxes with higher scores. However, due to the large number of prediction boxes, the prediction boxes generated in the regions with similar positions are predicting the same object, that is, these prediction boxes will overlap with each other. Then, if not processed, it will lead to one object being framed by multiple prediction boxes, and the visual effect is very poor. As shown in the following figure:

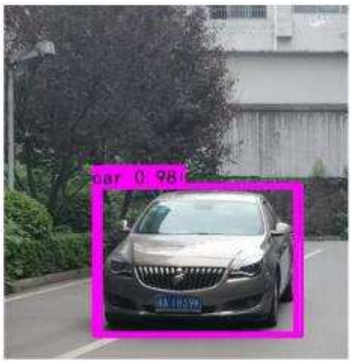

Fig. 14. Prediction effect without processing.

In order to display the prediction effect better, the Non-Maximum Suppression (NMS) algorithm is adopted here. Its main steps are: First, take the prediction box with the highest score among the prediction boxes, then calculate the IOU value between this prediction box and other prediction boxes, and find those prediction boxes whose IOU value is greater than the threshold (pre-set), indicating that these prediction boxes are predicting the same object, so delete them. Then, repeat the above steps continuously until the specified number of prediction boxes is found. The following figure is the detection effect diagram after the Non-Maximum Suppression algorithm:

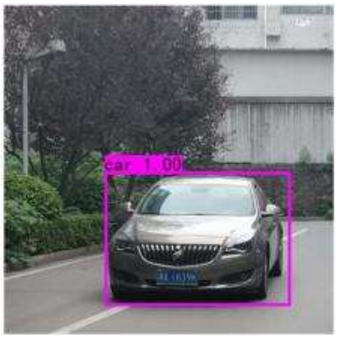

Fig. 15. The detection effect after non-maximum suppression.

V. DISPLAY OF SYSTEM OPERATION RESULTS

This system is based on YOLOv4 and improves the detection speed of the system by replacing the backbone feature extraction network, enhances the system's detection of small objects by improving the network structure, and improves the robustness of the system in the face of complex environments by expanding the dataset. Through a series of these improvement measures, a new model architecture is built, and finally a visual target detection and recognition processing system is formed. The prediction results generated by this system are presented below.

First, input the following picture into the system:

Fig. 16. Image to be detected.

Through visualization technology, the prediction results of the model are obtained, as shown in the figure:

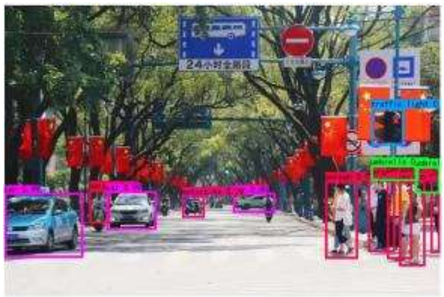

Fig. 17. Forecast result.

It can be seen from Figure 17 that the system has detected most of the objects to be detected in the picture, mainly including pedestrians and vehicles, and the detection accuracy of the system is also very high.

VI. CNCLUSION

The system focuses on the detection of pedestrians, vehicles and common objects in the road traffic environment. The core is that after the picture captured by the camera is input into the system, it can quickly and accurately identify the category information and location information of the objects to be detected. Due to the complex and changeable real road traffic environment, strict requirements are put forward for the system. It must have excellent detection speed to achieve real-time processing; have high accuracy to ensure accurate detection results; in the face of complex environments, it must have good robustness to ensure stable and reliable performance, thereby providing strong support for road traffic safety and management.

REFERENCES

[1] Mathe, S., Pirinen, A., & Sminchisescu, C. (2016). Reinforcement learning for visual object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2894-2902).

[2] Ke, W., Zhang, T., Huang, Z., Ye, Q., Liu, J., & Huang, D. (2020). Multiple anchor learning for visual object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10206-10215).

[3] Zhang, X., Wan, F., Liu, C., Ji, R., & Ye, Q. (2019). Freeanchor: Learning to match anchors for visual object detection. Advances in neural information processing systems, 32.

[4] Su, H., Deng, J., & Fei-Fei, L. (2012, July). Crowdsourcing annotations for visual object detection. In Workshops at the twenty-sixth AAAI conference on artificial intelligence.

[5] Amit, Y., Felzenszwalb, P., & Girshick, R. (2021). Object detection. In Computer Vision: A Reference Guide (pp. 875-883). Cham: Springer International Publishing.

[6] Wu, X., Sahoo, D., & Hoi, S. C. (2020). Recent advances in deep learning for object detection. Neurocomputing, 396, 39-64.

[7] Hussain, S. U., & Triggs, W. (2010, August). Feature sets and dimensionality reduction for visual object detection. In BMVC 2010-British Machine Vision Conference (pp. 112-1). BMVA Press.

[8] Torralba, A., Murphy, K. P., & Freeman, W. T. (2010). Using the forest to see the trees: exploiting context for visual object detection and localization. Communications of the ACM, 53(3), 107-114.

[9] Hara, K., Liu, M. Y., Tuzel, O., & Farahmand, A. M. (2017). Attentional network for visual object detection. arxiv preprint arxiv:1702.01478.

[10] Liu, Y., Huang, A., Luo, Y., Huang, H., Liu, Y., Chen, Y., ... & Yang, Q. (2020, April). Fedvision: An online visual object detection platform powered by federated learning. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 08, pp. 13172-13179).