№ 2-3(5), 2023 ISSN: 2534 – 8507 (print) ISSN: 2534 – 8515 (online)
INSTITUTE OF INFORMATICS AND INNOVATIVE TECHNOLOGIES

OBJECTS DETECTION IN AUTONOMOUS VEHICLES USING YOLOV5

Ngoc Dung Bui[1], Quang Tuyen Vu[1], Long Ngo[1], Thanh Binh Ngo[1], Dimitar Borisov[2]
University of Transport and Communications, Hanoi, Vietnam
University of Chemical Technology and Metallurgy, Sofia, Bulgaria

Abstract: **This paper presents a study on object detection technique specifically tailored for autonomous vehicles. Object detection plays a vital role in enabling autonomous vehicles to accurately perceive and understand their surroundings, ensuring safe and efficient navigation. Recently, YOLOv5 improves upon its predecessors by introducing advancements in architecture design, network scaling, and training techniques for object detection. This paper discusses the fundamental principles of YOLOv5, architecture and the concept of anchor boxes for bounding box prediction. Then, the training process, emphasizes the importance of a well-curated dataset and data augmentation techniques to improve model generalization. Experimental results showcasing the performance of YOLOv5 on benchmark object detection datasets, including its accuracy, speed, and real-time applicability demonstrate that YOLOv5 achieves state-of-the-art object detection results.**

Keywords: **object detection, autonomous vehicles, deep learning, yolo.**

I. INTRODUCTION

Object detection is a fundamental task in computer vision that involves identifying and localizing objects within an image or video [1]. It plays a crucial role in a wide range of applications, including autonomous driving, surveillance, robotics, and image understanding [2]. The ability to detect objects accurately and efficiently in real-world scenarios is essential for tasks such as object recognition, tracking, and scene understanding [3]. Over the years, significant advancements have been made in object detection algorithms, driven by the development of deep learning techniques and the availability of large-scale annotated datasets [4, 5]. These algorithms aim to overcome challenges such as object occlusion, variations in scale and pose, and cluttered backgrounds [6].

Deep learning approaches have revolutionized object detection, leading to breakthroughs in accuracy and efficiency. Convolutional Neural Networks (CNNs) have emerged as the backbone of many state-of-the-art object detection algorithms. They can automatically learn hierarchical representations from raw pixel data, enabling robust feature extraction and effective object recognition. Alexnet is one of the deep learning model that uses the ReLU activation function instead of the Tanh function to help archives high accuracy in object recognition [7]. The series of Fast RCNN and Faster RCNN with the Region Proposal Network (RPN) improve the accuracy and speed of the object detection algorithm [8, 9]. An RPN takes an image of any size as input and outputs a region proposal (set of locations of rectangles that can contain objects), along with the corresponding rectangle's probability of containing the object. Single Shot Multibox Detector (SSD) based on the VGG16 employs the feature layer to detect small objects [10]. YOLO (you only look once) series improve the speed of object detection, for example, YOLOV3 uses darknet-53 feature extraction to minimize the parameters of the model [11].

Detecting target objects on the road is an essential task for autonomous driving. For most existing road object detectors, the detection accuracy for small objects is less than half that of large objects [12]. The reason is that they usually cover fewer pixels, and it is difficult to extract features from low resolution, so the model can easily confuse it with the background, resulting in missed or incorrect detection. In this paper, we present a comprehensive study of the YOLOv5 algorithm for object detection in the context of autonomous vehicles. We aim to analyze the key features and improvements introduced in YOLOv5, assess its detection performance, and compare it with other state-of-the-art object detection algorithms. We also explore the impact of various factors, such as network architecture, training strategies, and dataset characteristics, on the performance of YOLOv5 in autonomous driving scenarios.

The remainder of this paper is organized as follows. Section 2 presents the YOLO architecture and applications of YOLOV5 in object detection and autonomous vehicles. Section 3 shows the empirical results of our approach, while Section 4 concludes the paper.

II. APPLICATION OF YOLOV5 IN OBJECT DETECTION

A. YOLO

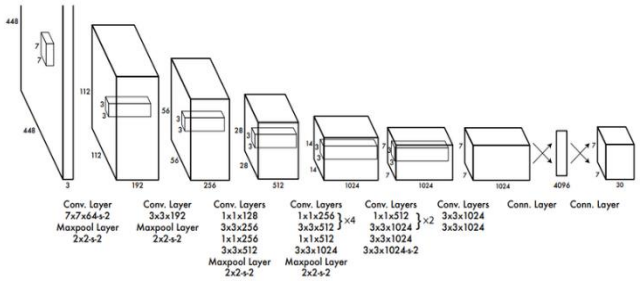

Fig. 1. The YOLO architecture

YOLO (You Only Look Once) is an object detection algorithm that has gained popularity due to its real-time performance and accuracy [13]. YOLO approaches object detection as a regression problem, where it predicts bounding boxes and class probabilities directly from an input image, all in one pass. This differs from traditional object detection methods that rely on region proposal algorithms followed by classification. The YOLO algorithm operates on a grid system. The input image is divided into an S x S grid, and each grid cell is responsible for predicting bounding boxes for objects that fall within its boundaries. For each grid cell, YOLO predicts B bounding boxes and the corresponding class probabilities. The class probabilities represent the likelihood of each object class being present within the bounding box. YOLO also predicts the confidence score for each bounding box, which reflects the accuracy of the predicted box. The confidence score is a combination of the objectless score (probability of an object being present in the box) and the accuracy of the predicted box. During training, YOLO uses labeled bounding box annotations to compute the loss function, which consists of multiple components. The loss function penalizes incorrect predictions of object locations and class probabilities, aiming to minimize the discrepancy between the predicted values and ground truth. The total loss function is optimized by the sum-squared error in the output as equation 1:

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left[\left(x_i-\hat{x}_i\right)^2+\left(y_i-\hat{y}_i\right)^2\right]$$
$$+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left[\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right]$$
$$+\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left(C_i-\hat{C}_i\right)^2+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{noobj}\left(C_i-\hat{C}_i\right)^2$$
$$+\sum_{i=0}^{S^2}1_{ij}^{obj}\sum_{c\in classes}\left(p_i(c)-\hat{p}_i(c)\right)^2$$

To improve localization accuracy, YOLO uses anchor boxes. These anchor boxes are pre-defined bounding boxes of different sizes and aspect ratios, which are used to predict more accurate bounding box coordinates relative to the anchor boxes. Once trained, YOLO can be used for object detection on new images. The algorithm takes an input image, processes it through a convolutional neural network (usually based on a pre-trained architecture like Darknet), and applies the YOLO algorithm to generate bounding box predictions and class probabilities for the objects present in the image. The architecture of YOLO is shown in figure 1. However, each grid cell only predicts two boxes and can only have one class, therefore it limits the prediction of the nearby objects.

B. Object detection using YOLOV5

YOLOv5 offers several advantages that make it a preferred choice for object detection tasks. It combines speed and efficiency, enabling real-time inference without compromising accuracy. The flexibility and customization options allow users to adapt the model to specific detection requirements, while transfer learning capabilities reduce training time and resource requirements. With support for various object classes and the availability of pre-trained models, YOLOv5 provides a versatile solution that can handle complex scenes and achieve good performance even with limited data. Its active development, cross-platform compatibility, and community support ensure ongoing improvements and compatibility with the latest advancements in deep learning frameworks, making YOLOv5 an excellent option for a wide range of object detection applications. Figure 2 show the YOLOv5 architecture [14].

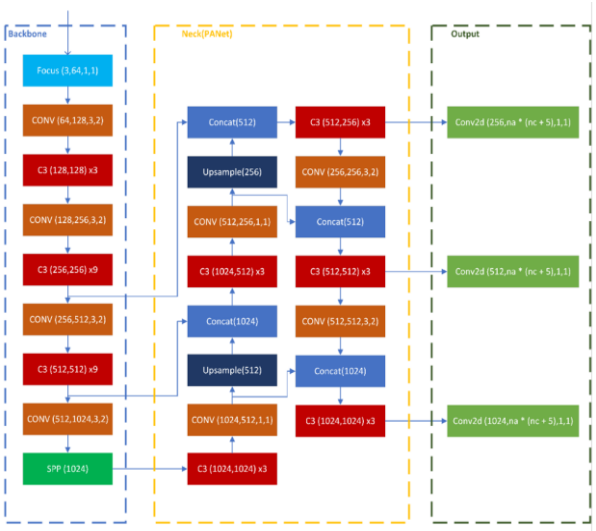

Fig. 2. YOLOv5 architecture

YOLOv5 returns three outputs: the classes of the detected objects, their bounding boxes and the objectness scores. Thus, it uses BCE (Binary Cross Entropy) to compute the classes loss and the objectness loss. While CIoU (Complete Intersection over Union) loss to compute the location loss. The formula for the final loss is given by the equation 2.

$$Loss = \lambda_1 L_{cls} + \lambda_2 L_{obj} + \lambda_3 L_{loc} \quad (2)$$

Object detection using YOLOv5 is an efficient and powerful approach that allows for accurate and real-time detection of objects in images and videos, suitable in autonomous vehicles. The key steps involved in object detection using YOLOv5 are as follows:

- Data Preparation: Prepare a labeled dataset with annotated bounding boxes around the objects of interest. The dataset should contain images or videos along with their corresponding object labels.

- Model Architecture: YOLOv5 utilizes a deep convolutional neural network (CNN) architecture. It consists of multiple convolutional layers, followed by upsampling and downsampling layers, which allow the model to detect objects at various scales as in figure 2.

- Model Training: Train the YOLOv5 model using the prepared dataset. The training process involves optimizing the model's parameters using techniques like backpropagation and gradient descent. YOLOv5 implements advanced training strategies such as focal loss, mosaic data augmentation, and transfer learning to improve detection accuracy.

- Inference: Once the model is trained, it can be used for object detection on new images or videos. During inference, the model processes the input data and generates bounding boxes around the detected objects, along with their corresponding class labels and confidence scores.

- Post-processing: Apply post-processing techniques to refine the detection results. This may involve filtering out low-confidence detections, performing non-maximum suppression to remove redundant bounding boxes, and applying thresholds for confidence scores.

Object detection step plays a crucial role in the context of autonomous vehicles, providing a foundational capability for understanding the surrounding environment. By employing sophisticated computer vision techniques and deep learning models, object detection enables autonomous vehicles to accurately identify and locate various objects, including vehicles, pedestrians, cyclists, traffic signs, and obstacles. This information is vital for critical tasks such as collision avoidance, path planning, and decision-making, ensuring the safe and efficient operation of autonomous vehicles. Object detection in autonomous vehicles combines real-time processing, high accuracy, and robustness to handle diverse and challenging scenarios on the road, making it an essential component for achieving reliable and trustworthy autonomous driving systems.

III. Experimental Results

A. Dataset

To train our system, we use dataset includes 15000 images with 97.942 labels across 11 classes of pedestrians, bikers, cars, and traffic lights. All images are 1920x1200 size. The label for the dataset is shown in table 1 [15]. Figure 3 shows the sample of the dataset.

Table. 1. Class balance of the dataset

Class	Number
Car	64.399
Pedestrian	10.806
trafficLight-Red	6.870
trafficLight-Green	5.465
Truck	3.623
trafficLight	2.568
Biker	1.864

trafficLight-RedLeft	1.751
trafficLight-GreenLeft	310
trafficLight-Yellow	272
trafficLight-YellowLeft	14

The samples of images in the dataset are shown in figure 3. In the dataset, all the images are captured using the camera in the moving vehicles. These images are labeling and give as input to train the model. Here the output of the model will be 11 classes corresponding to the table 1.

Fig. 3. Images sample in the dataset

B. Experimental results

Data are separated into training and testing set randomly. The training part has a part of 80% of all the data, the others are for testing. During training process, the features and labels are provided for all the images in the training data set. The goal is to capture the relationship between features and class labels. The performance of the method is shown in Fig. 4.

Figure 4 shows the precision-recall curve that described the performance of the classification model. Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive. It quantifies the model's ability to correctly identify true positives while minimizing false positives. Recall, on the other hand, calculates the proportion of correctly predicted positive instances out of all actual positive instances. It represents the model's ability to capture all positive instances while minimizing false negatives. To construct a precision-recall curve, the classification model is first applied to a set of test data. Then, different decision thresholds are applied to generate a series of precision and recall values. These values are plotted on a graph, with recall on the x-axis and precision on the y-axis. Each point on the curve corresponds to a specific decision threshold.

$$P = \frac{T_p}{T_p + F_p}, R = \frac{T_p}{T_p + F_n} \quad (3)$$

Where Tp is true positive, Fp is false positive, Fn is false negative.

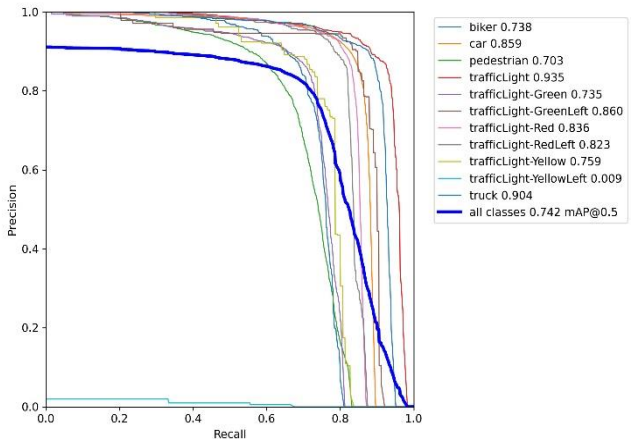

Fig. 4. Performance of the model

From figure 4, the accuracy of the model is 0.7. We can see that the accuracy of the test set is linear with the size of the class dataset. The trafficLight-Yellow got lowest accuracy due to the low number of the samples. The visualization of the objects detection is shown in figure 5.

Fig. 5. Multiple objects detection

CONCLUSIONS

In this paper, we presented the multiple objects detection based on YOLOv5. YOLOv5 combines speed, accuracy, and efficiency, allowing real-time detection of various objects in complex and dynamic driving scenarios. Its deep learning-based approach, leveraging convolutional neural networks, enables accurate identification and tracking of objects, including vehicles, pedestrians, and traffic signs. The transfer learning capabilities of YOLOv5 reduce training time and resource requirements, making it adaptable to different detection requirements and datasets. Additionally, YOLOv5's active development, cross-platform compatibility, and strong community support ensure ongoing improvements and compatibility with the latest advancements in deep learning frameworks.

ACKNOWLEDGEMENTS

This research is funded by the University of Transport and Communications (UTC) under grant number T2023-CN-KDN-002.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need. Advances in neural information processing systems, 30, 2017.

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate". 3rd International Conference on Learning Representations, 2015.

[3] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units". Proceedings of the 54th annual meeting of the association for computational linguistics, 2016.

[4] X. Li, H. Yan, X. Qiu, and X. Huang, "FLAT: Chinese NER using flat-lattice transformer". arXiv preprint arXiv:2004.11795, 2020.

[5] D. Q. Nguyen, and A. T. Nguyen, "PhoBERT: Pre-trained language models for Vietnamese". arXiv preprint arXiv:2003.00744, 2020.

[6] T. Poibeau. Machine translation. MIT Press, 2017.

[7] Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 2012, 25, 1097–1105.

[8] Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.

[9] Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell. 2017, 39, 1137–1149.

[10] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single shot multi-box detector. In European Conference on Computer Vision; Springer: Cham, Switzerland, 2016; pp. 21–37.

[11] Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. arXiv 2018, arXiv:1804.027671.

[12] Bharat Mahaur, K.K. Mishra, Small-object detection based on YOLOv5 in autonomous driving systems, Pattern Recognition Letters, Vol. 168, pp. 115-122, 2023.

[13] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, You Only Look Once: Unified, Real-Time Object Detection, arXiv:1506.02640v, 2016.

[14] Github: Yolov5. https://github.com/ultralytics/yolov5.

[15] roboflow, https://public.roboflow.com/object-detection/self-driving-car , MIT, 2022