Vehicle And Pedestrian Detection Algorithm Based on Improved YOLOv5

Jiuhan Sun, Zhifeng Wang

***Abstract*—As urbanization progresses, urban road congestion has intensified, highlighting the need for effective vehicle and pedestrian detection as a cornerstone of public safety transportation. This area holds significant relevance in video surveillance and public safety domains. Despite its importance, achieving precise vehicle and pedestrian detection in complex road environments remains challenging. This paper presents a vehicle-pedestrian detection algorithm based on the improved YOLOv5. Key modifications include the integration of a small target detection layer and alterations to the feature pyramid using the feature fusion technique inherent to the weighted Bi-directional Feature Pyramid Network (BIFPN). This ensures efficient multi-scale feature fusion. A coordinated attention mechanism is introduced to preserve accurate target location data. Furthermore, the paper incorporates the SIOU metric to refine the localization loss function, bolstering both speed and edge regression accuracy. Experimental outcomes indicate that our improved YOLOv5 algorithm augments detection accuracy by 1.9% and achieves a detection speed of 67 FPS, which surpasses many competing target detection algorithms.**

***Index Terms*—KITTI, Feature fusion, Attention mechanism, SIoU loss function,YOLOv5.**

I. Introduction

VEHICLE and pedestrian detection holds pivotal importance in the realms of intelligent transportation and autonomous vehicles. As modern technology has advanced, vehicles have transitioned from merely assisted driving to capabilities of full autonomous operation. Within the intricate urban traffic environment, vehicles and pedestrians are two key elements. For safety considerations, intelligent vehicles employ on-board cameras to gather road data [1]. Subsequently, target detection technologies autonomously and accurately identify vehicles and pedestrians, pinpointing their locations and highlighting them with predictive frames. This provides real-time feedback and early warnings, enhancing the driver's awareness of the traffic surroundings and improving the vehicle's obstacle avoidance during autonomous operations. Such advancements not only fortify the safety of vehicular movement and pedestrian crossings but also play a crucial role in mitigating traffic incidents. Hence, the research and refinement of vehicle and pedestrian detection algorithms bear significant relevance.

The crux of target detection hinges on feature extraction. Broadly, the methodologies can be bifurcated into traditional target detection algorithms and those rooted in deep learning, based on their feature extraction techniques. Traditional object detection algorithms first process an image, followed by a sliding window traversal over the input image. This operation builds targets employing manually designed features such as HOG [2] and Haar [3]. Despite their foundational importance, these traditional techniques, reliant on manually designed features, often suffer from limited robustness and weak feature representation, leading to suboptimal outcomes. With the advent of deep learning, achieving breakthroughs across myriad domains, target detection underwent significant transformations. Deep learning-based target detection algorithms primarily fall into two categories. The first is two-stage target detection algorithms, which are exemplified by R-CNN [4], Fast R-CNN [5], and Faster R-CNN [6]. These algorithms typically commence by generating candidate regions that might encompass targets, using candidate region networks. Subsequent steps involve neural networks extracting features from these regions, culminating in target classification and location regression. While these algorithms boast commendable accuracy, their need to extract and process an extensive array of candidate regions translates to protracted computation times and diminished efficiency. The other is known as one-stage target detection algorithms. Eschewing the extraction of candidate regions, these algorithms directly feed images into networks for feature extraction, followed by bounding box regression. YOLO [7] and SSD [8] stand out as archetypal representations. While this class of algorithm excels in detection speed, it falls short of being optimal for target detection purposes.

In this paper, we present an enhanced YOLOv5 algorithm specifically tailored for vehicle and pedestrian detection. Our primary contributions include the design of a multi-scale feature fusion module, which adds a small target detection layer atop the original three-scale layer and integrates a jump connection between the backbone network and output layers. This refines the existing feature pyramid structure, improving both detection and feature fusion capabilities. To account for the relationship between channels and positions, we've embedded a coordinate attention mechanism, allowing the model to pinpoint targets with greater accuracy. Moreover, we've adopted SIOU in place of CIOU in the loss function, addressing issues of reduced detection accuracy and excessive background interference, which resulted in more accurate prediction outcomes.

II. Related Studies

As deep learning techniques continue to evolve, deep neural networks have emerged as the predominant methodology in target detection. The Convolutional Neural Network (CNN), in particular, has gained immense popularity due

Manuscript received May 10, 2023; revised September 6, 2023.

This work was supported by the National Natural Science Foundation of China (61575090, 61775169), the Natural Science Foundation of Liaoning Province (2019-ZD-0267) and the Liaoning Provincial Education Department (2020LNJC01).

Jiuhan Sun is a postgraduate student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China. (e-mail: 188527474@qq.com)

Zhifeng Wang is an associate Professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China. (corresponding author to provide phone: 150-4234-1839; e-mail: wangzhifeng_sia@126.com).

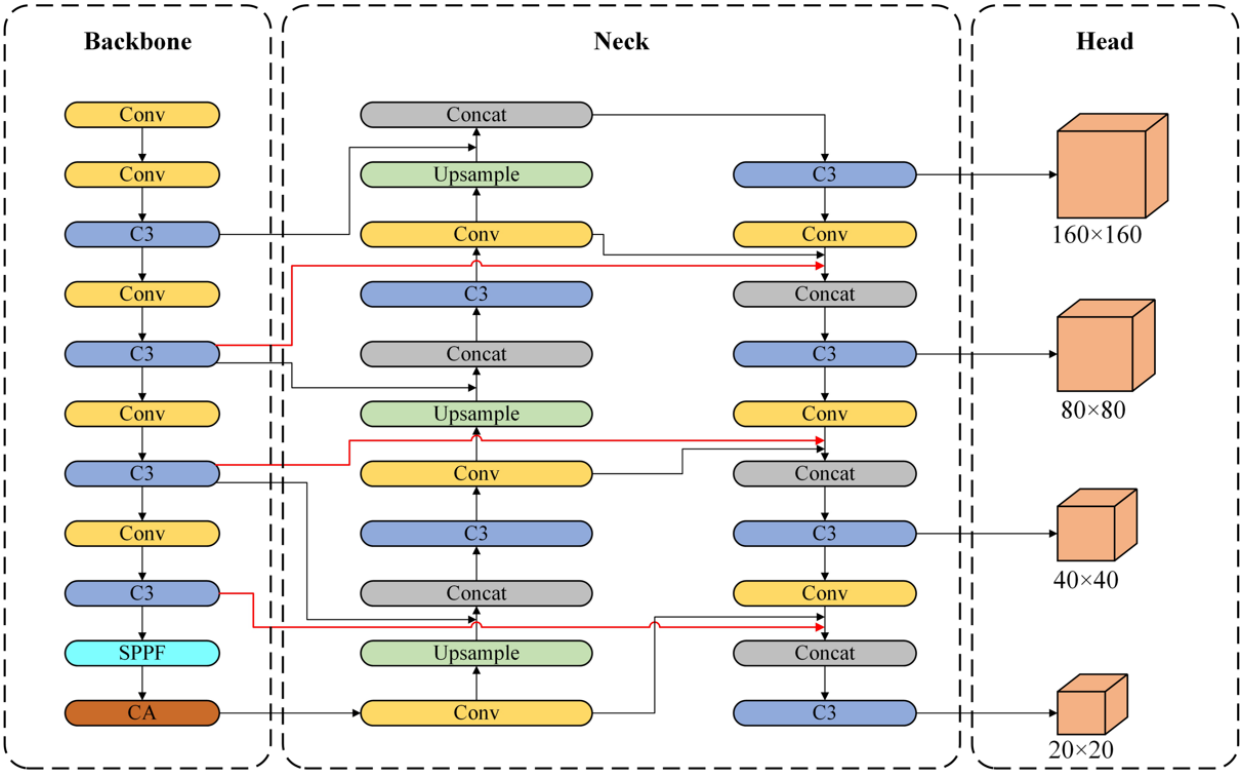


Fig. 1: Improved YOLOv5 network aechitecture

to its superiority over traditional algorithms in image data processing and its capacity to autonomously extract and learn features. This makes it highly applicable to vehicle and pedestrian detection. Cao et al. [9] introduced a multi-branched advanced network. This network leverages branches of varying resolutions and perceptual fields to extract sophisticated semantic features. By cross-linking layers and employing Atrous Convolution, they augmented the resolution of the feature map, thereby enhancing spatial information crucial for pinpointing small-target pedestrians. In a different vein, Nguyen et al. [10] unveiled an enhanced framework rooted in Faster-RCNN. By deploying Depth-Separable convolution in crafting convolutional layers and integrating a context-aware pooling layer, targets are resized to specific dimensions without compromising vital contextual data. This technique outstrips several traditional target detection algorithms in performance metrics. Chen et al. [11] put forth an optimized SSD algorithm, tailored for rapid vehicle detection in traffic scenarios. This algorithm employs MobileNetV2 as its foundational network and incorporates a channel attention mechanism for feature weighting. By utilizing a deconvolution module to establish a feature fusion structure, the method achieved a commendable 84.83% accuracy on the KITTI dataset. Guo et al. [12], while adopting the SSD model with ResNet50 as the backbone, integrated an attention mechanism. Notably, while this approach elevated detection accuracy, it encountered challenges with multiple vehicle targets and pronounced occlusion scenarios. The fusion of diverse output size feature maps, combined with the addition of convolutional layers, escalated the model's complexity, rendering it unsuitable for real-time applications. Yuan et al. [13] presented an enhanced YOLOv4-based vehicle detection algorithm. This modification replaces YOLOv4's original backbone network, CSPDarknet53, with MobileNetv3 and substitutes 3×3 convolution in the original network's feature extraction segment with DepthSeparable convolution. By innovatively redesigning the loss function using a weighting technique, the refined approach amplifies accuracy by 0.53% and curtails the model parameters by a staggering 78%.

While existing methodologies have made notable strides in vehicle and pedestrian detection, the dynamic and evolving nature of traffic scenarios necessitates continual exploration of innovative techniques and approaches. Given the intricate nature of actual road traffic scenarios, many prevalent target detection algorithms struggle to accurately identify small-sized targets. Complications such as mutual occlusions further compromise the model's detection precision, resulting in issues like false detections and missed targets. Consequently, this accentuates the need for more sophisticated detection algorithms. This study endeavors to holistically address these challenges—specifically the accurate detection of smaller targets and managing occlusions—whilst ensuring real-time responsiveness. Our ultimate aim is to strike an optimal balance between detection precision and processing speed.

III. YOLOv5 ALGORITHM PRINCIPLE

YOLOv5 stands out as a single-stage target detection algorithm, boasting faster training and inference times than its predecessor, YOLOv4, alongside marked enhancements in detection speed and efficiency. The YOLOv5 models are categorized into five versions based on their weight sizes: YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, with each subsequent model having increased width and depth. For our study, we opted for YOLOv5s due to its compact weight file, offering the fastest processing speed and minimal computational resource consumption

among the variants. Structurally, the model comprises three primary components: the Backbone network, the Neck feature fusion layer, and the Head output layer, as illustrated in Fig. 2. These parts collaboratively ensure optimal target detection efficacy.

The backbone network primarily comprises three components: the feature extraction module (C3), the convolution module (Conv), and the spatial pyramid pooling (SPPF). This network encompasses five standard convolutional layers, labeled as Conv, which are instrumental in extracting features from the input image. Within the C3 module, the input

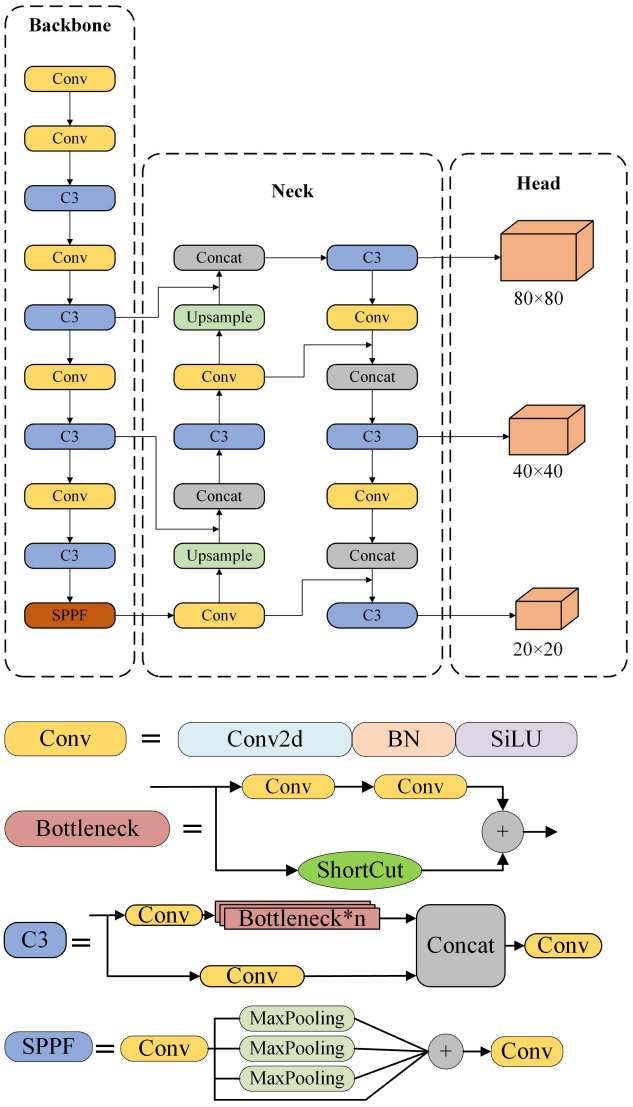


Fig. 2: YOLOv5 network aechitecture

feature map is channeled through two distinct branches. The first branch is processed through a Conv module before undergoing feature learning via a stacked Bottleneck module; the second branch serves as a residual connection and solely passes through a Conv module. Upon processing, these branches are channel-wise concatenated and subsequently output through another Conv module. After each convolution, the C3 module performs feature stacking, which enriches the model's representational capability. Positioned at the backbone network's terminal feature layer is the SPPF structure. Here, input feature maps undergo convolution before being sequentially processed by three maximum pooling layers, each sized 5 × 5. This arrangement amplifies the model's perceptual prowess.

The Neck layer is comprised of both a Feature Pyramid Network (FPN) and a Path Aggregation Network (PAN). The FPN's top-down feature pyramid is integrated with the PAN's bottom-up feature pyramid. This integration facilitates the fusion of features from different stages of the feature maps, minimizing information loss. As a result, the model's proficiency in detecting targets across varying scales is significantly bolstered.

The Head output layer initiates its process by performing convolution operations on the 20×20, 40×40, and 80×80 feature maps, which are the outputs from the Neck layer. These operations are vital for detecting the class and location of the target. Additionally, the non-maximum suppression algorithm is employed to select among the prediction frames from all detection layers. The frame with the utmost confidence is retained, culminating the entire detection process.

IV. IMPROVEMENT STRATEGY

A. *Multi-scale enhanced feature fusion*

The original YOLOv5 network model conducts detection across three different scales of feature maps, allocating three distinct detection frames to each scale. By doing so, it aims to detect large, medium, and small targets using the scales 20×20, 40×40, and 80×80. However, real roads often present numerous small, distant targets. Given that the perceptual field of the 80×80 detection layer in the original model is limited to 8×8, the model struggles to identify targets in the original image if their size is less than 8 pixels. Consequently, the original 80×80 detection layer, designed for small targets, often misses these even smaller targets, reducing the YOLOv5 network's detection efficacy.

To enhance the YOLOv5s' detection capability for small targets, we incorporated an additional 160×160 scale layer specifically designed for detecting these diminutive entities. Originating from the backbone network, this new branch upsamples the 80×80 feature map to facilitate feature fusion. Fig. 3 offers a schematic representation of this multiscale detection. By amalgamating deep and shallow insights, the model enriches its expression of the semantic features and spatial data pertaining to small targets. The refined feature layer is then forwarded to the newly introduced detection head for decoding. Consequently, the original three-scale detection has evolved into a four-scale system, with final output feature maps sized at 160×160, 80×80, 40×40, and 20×20. This expansion in the detectable size range equips our model to more adeptly identify small distant targets, leading to an improved overall detection outcome.

YOLOv5s employs the structure of FPN [14], illustrated in Fig. 4(a). This configuration forms a top-down channel fusion to seamlessly integrate the image semantic layer with its feature layer. However, it is hindered by a unidirectional flow of information. To counter this limitation, the PAN structure, showcased in Fig. 4(b), [15] is introduced. By integrating a bottom-up channel to the existing FPN, it ensures the prediction feature layer imbibes both upper and lower semantic information. Building on PAN, the BIFPN structure optimizes the network by removing less contributive nodes and introducing new jump connections between nodes

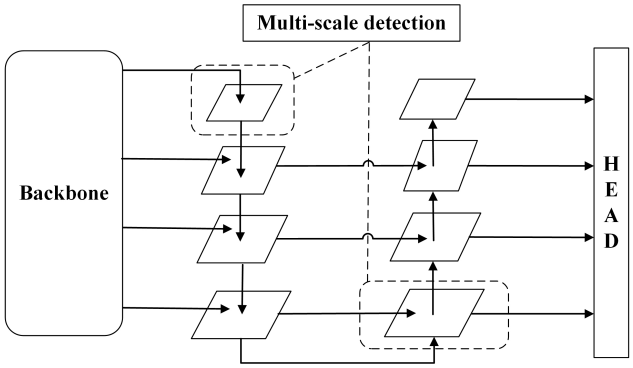


Fig. 3: Multi-scale extended structure diagram

at equivalent levels [16]. As shown in the Fig. 4(c), P3-P7 signify feature maps across varied layers, with P4-P6 as intermediates. The blue arrows highlight top-down semantic flow, the red arrows show bottom-up location flow, and purple arrows signify new pathways linking input-output nodes. This BIFPN arrangement interprets each bidirectional path as a distinct feature layer, facilitating fusion of information from varying layer feature maps. Such a mechanism reduces resource usage, effectively manages image noise and disturbances, and even assigns weights based on input feature significance, normalizing them within a [0, 1] range.

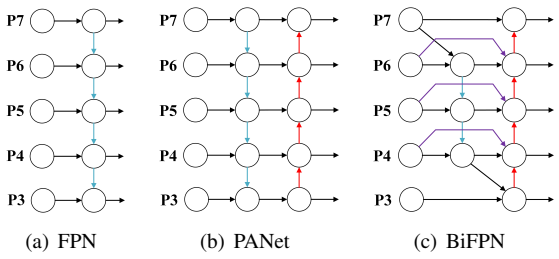


Fig. 4: Feature fusion method.

Drawing inspiration from BIFPN, this study incorporates its feature fusion approach while omitting the weight aspect, aiming to elevate the model's feature integration and target recognition capabilities. In the YOLOv5's PAN structure, we integrated three novel feature fusion pathways, depicted by the red line in Fig. 1. This ensures each node within the fusion architecture maintains a connection to the original feature layer stemming from the backbone network, fostering consistent engagement with the original feature data and enhancing inter-part feature fusion. Such modifications not only heighten the model's sensitivity to various target scenarios but also bolster detection accuracy without significantly escalating computational demands. Consequently, this optimized model is more adept at tackling intricate target detection challenges.

B. Attentional mechanisms

At its core, the attention mechanism seeks to emulate human perception and attention within machines. By integrating this mechanism, neural networks learn to prioritize essential data pertinent to the recognition task while filtering out irrelevant information, such as background elements. This strategic focus subsequently enhances the overall performance of the model.

To bolster the model's detection capabilities, attention mechanisms have become increasingly integrated into target detection. Prominent mechanisms like the Squeeze and Excite attention (SE) [17], the former focusing primarily on channel information while often overlooking spatial structures, and the latter, the Convolutional Block Attention Module (CBAM) [18], amalgamating spatial and channel information by using global pooling, but capturing only partial location details. In this study, we introduce a coordinated attention mechanism post-SPPF. This approach not only uncovers more nuanced, distance-dependent correlations but also assimilates the target's spatial structure. Consequently, it amplifies the model's feature learning prowess, steering it towards more pertinent information for enhanced target recognition.

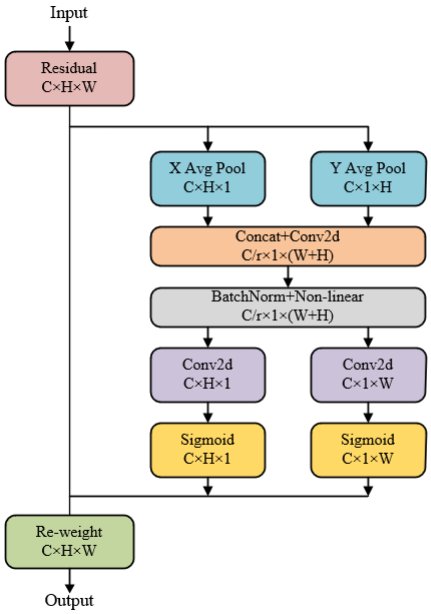


Fig. 5: Coordinate attention mechanisms

The coordinated attention mechanism, depicted in Fig. 5, represents an innovative attention module. Central to its design is the incorporation of candidate frame location information into channel attention, facilitating rapid pinpointing of interest areas. This mitigates the need for two-dimensional global pooling which typically reduces a feature tensor to a single vector, leading to location data loss [19]. Specifically, the mechanism averages all input feature map channels along the horizontal and vertical coordinates, yielding C×H×1 and C×1×W feature maps. After concat operations and 1×1 convolution, results are split into horizontal and vertical tensors for further 1×1 convolution, followed by Sigmoid activation. This multiplies with the input feature map, optimizing model detection without raising computational demands and preserving both location and cross-channel information.

C. Optimization of loss function

Non-maximum suppression (NMS) is a prevalent post-processing technique in object detection. Essentially, NMS is a method of iteratively refining the set of detection candidates. Initially, each candidate bounding box, along with its

associated classification score, is generated by the classifier. These scores are then sorted, and a confidence threshold for the bounding boxes is established. The Intersection over Union (IoU) between every box and the one with the highest score is computed. If the IoU exceeds a certain threshold, the corresponding candidate box is removed. IoU is a widely-used metric in object detection. Its purpose is to assess the accuracy of the predicted bounding box's location compared to the actual location. Essentially, it calculates the overlap between the predicted and ground truth bounding boxes. This is done by dividing the intersection area of the predicted and true boxes by their union area. The formula for IoU can be found in Fig. 6.

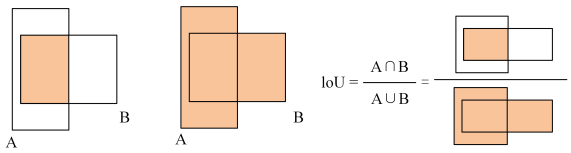


Fig. 6: IOU calculation method

The loss function quantifies the discrepancy between a model's predicted frame and the actual frame, playing a pivotal role in shaping the model's performance. In YOLOv5, there are three different loss functions: category loss, confidence loss, and localization loss. Within the localization loss, CIOU is employed as the loss function to measure the similarity between the predicted frame and the true frame. However, CIOU does not account for the orientation mismatch between the predicted and actual frames, leading to a slower convergence during model training. To rectify this, our study introduces the SIOU loss function [20], as depicted in Fig. 7. Superior to CIOU, SIOU not only factors in overlap, frame distance, and aspect ratio but also integrates the vector angle between actual and predicted frames. This refined approach not only accelerates training but also elevates inference accuracy. Specifically, SIOU amalgamates four components: angle loss, distance loss, shape loss, and IOU loss, with the respective equations presented in Equations (1) through (4).

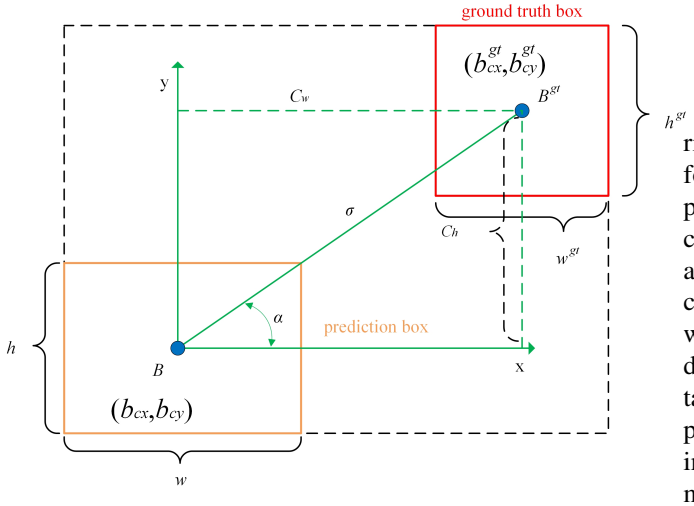


Fig. 7: Schematic diagram of angular costing

$$\Lambda = 1 - 2 * \sin^2(arcsin(\frac{c_h}{\sigma} - \frac{\pi}{4})) \tag{1}$$

In the equation above, σ denotes the distance between the center points of the true bounding box and the predicted one, and ch represents the height difference between the center points of these two boxes.

$$\Delta = \sum_{t=x,y} (1 - e^{-\gamma\rho t}) \tag{2}$$

$$\rho_x = (\frac{b_{c_x}^{gt} - b_{c_x}}{c_w}) \tag{3}$$

$$\rho_y = (\frac{b_{c_y}^{gt} - b_{c_y}}{c_h}) \tag{4}$$

In the equations, c_w and c_h refer to the width and height, respectively, of the smallest enclosing rectangle formed by the centers of the actual and predicted frames. The parameter "γ" is given by $\gamma = 2 - \Lambda$, here $b_{c_x}^{gt}$ and $b_{c_y}^{gt}$ represent the coordinates of the center of the actual frame, while b_{c_x} and b_{c_y} denote the coordinates of the center of the predicted frame.

The shape loss quantifies the discrepancy between the central positions of the predicted and actual frames. Its purpose is to achieve the most accurate prediction frame. The specifics of this deviation are elaborated in Equations (5), (6), and (7).

$$\Omega = \sum_{t=w,h} \left(1 - e^{-w_t}\right)^{\theta} \tag{5}$$

$$w_w = \frac{|w - w^{gt}|}{\max(w, w^{gt})} \tag{6}$$

$$w_h = \frac{|h - h^{gt}|}{\max(h, h^{gt})} \tag{7}$$

Where, w, h, w_{gt}, h_{gt} refer to the width and height of the predicted and real frames, respectively, and θ controls the degree of attention to shape loss. To summarize, the calculation of the SIOU loss function is illustrated by Equation (8).

$$Loss_{SIOU} = 1 - IOU + \frac{\Delta + \Omega}{2} \tag{8}$$

V. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we evaluate the enhanced YOLOv5 algorithm. We begin by detailing the dataset selection process for our experiment and outlining the configuration of experimental parameters. Subsequently, we compare the P-R curves of the unimproved and improved YOLOv5 models, allowing us to quantify the detection accuracy for each category during the model evaluation phase. Moving forward, we conduct a comparative experiment on the KITTI dataset, juxtaposing the improved model against different target detection algorithms, and subsequently assessing its performance. Concluding our evaluation, we input select images from the test set into YOLOv5 using our enhanced model, thereby facilitating a direct comparison of detection outcomes. Additionally, we perform ablation experiments on the enhanced model, providing insights into the effects of integrating distinct modules on the model's detection performance.

A. Dataset Selection

In this experiment, we opted to utilize the publicly available KITTI dataset both as our training and testing samples. Renowned as one of the largest international datasets, KITTI offers an invaluable resource for evaluating computer vision algorithms in autonomous driving scenarios. The dataset's images encompass a diverse array of challenges, ranging from a substantial count of diminutive targets to varying degrees of occlusion, thereby presenting a formidable testbed for prevailing target detection algorithms. To facilitate compatibility with YOLO format, we undertook the conversion of the KITTI dataset into a text-tagged file, while also refining the Labels tag structure into two primary classes. This reclassification involved merging 'Car,' 'Van,' 'Truck,' and 'Tram' into the 'Car' class, and consolidating 'Pedestrian,' 'Person sitting,' and 'Cyclist' into the 'Person' class. Concurrently, we omitted the 'Misc' and 'DontCare' classes. Within our dataset, we allocated 7,491 images, distributing them into a training set and a test set with a 9:1 ratio. To offer visual context, a subset of sample images from the experimental dataset can be observed in Fig. 8.

Fig. 8: An image from a portion of an experimental dataset

Fig. 9 illustrates the distribution of label sizes across all categories within the dataset. The horizontal axis represents the width of the label box, while the vertical axis corresponds to the height of the label box. It is evident that a concentration of points is observed in the lower left corner of the graph. This clustering signifies a substantial prevalence of small targets within the KITTI dataset, aligning cohesively with the primary focus of our research problem addressed in this paper.

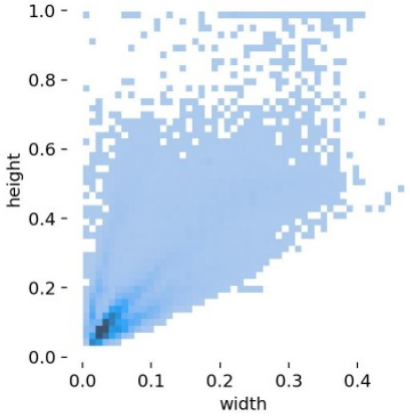
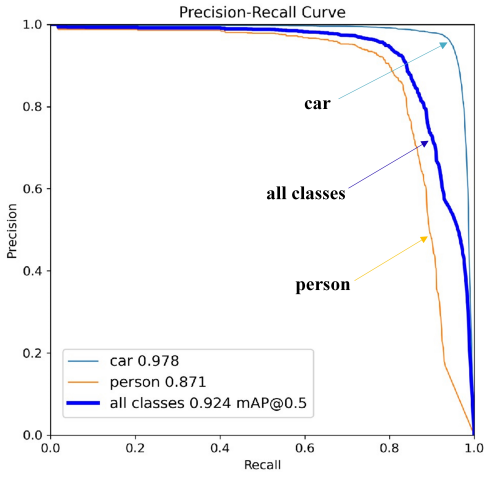


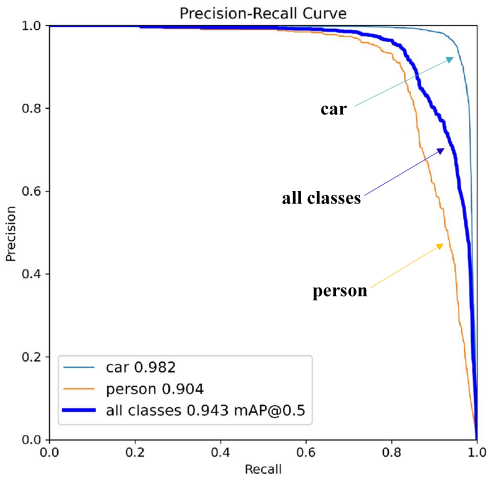
Fig. 9: Schematic diagram of the distribution location of labels in the KITTI dataset

B. Experimental Results and Analysis

The experimental framework within this paper encompasses both a hardware and a software platform. The hardware setup was established on an NVIDIA GeForce GTX 3090, operating within the CUDA 10.2 environment. The model training process was expedited through GPU acceleration. The software platform, on the other hand, is anchored in the PyTorch 1.9.0 deep learning framework, facilitated by the Windows 10 operating system, and executed within the PyCharm community IDE. We employed pre-trained weights from the COCO dataset as the initial weights. To mitigate the risk of the model converging to a local optimum solution, the Stochastic Gradient Descent (SGD) optimizer was harnessed to iteratively update the network parameters. In this context, the model's initial learning rate was configured at 0.01, and the momentum parameter was set to 0.937. Each training round incorporated a batch size of 32, spanning 200 rounds.



(a) P-R curve of the YOLOv5 algorithm



(b) P-R curve of the improved YOLOv5 algorithm

Fig. 10: Comparison of P–R curves of two models.

The core focus of this experiment pertains to the evaluation of the algorithm's recognition speed and recognition capability. The assessment of recognition speed is based on the number of images recognized per second, quantified as Frames Per Second (FPS). Meanwhile, recognition ability

is gauged through selected metrics, including accuracy (P), recall (R), and mean Average Precision (mAP). Accuracy serves as a measure of the model's detection precision, whereas recall appraises the comprehensiveness of model detection, and mAP, encapsulating both accuracy and recall, furnishes an amalgamated perspective. Its value positively correlates with model detection efficacy. The formulations of these metrics are as follows:

$$P = \frac{TP}{TP + FP} \tag{9}$$

$$R = \frac{TP}{TP + FN} \tag{10}$$

$$AP = \int_0^1 P(R)dR \tag{11}$$

$$mAP = \frac{1}{n}\sum_{i=0}^{n} AP_i \tag{12}$$

Here, the notation breakdown is as follows: 'TP' denotes the count of positive samples correctly identified as positive classes, 'FP' represents the count of negative samples erroneously classified as positive classes, 'FN' signifies the count of positive samples incorrectly recognized as negative classes, and 'n' symbolizes the total number of classes within the dataset. 'AP' corresponds to the average accuracy within the same category, and the area under the Precision-Recall (P-R) curve, delineated by accuracy 'P' and recall 'R', determines the 'AP' value. 'mAP' stands for the mean Average Precision, denoting the average of the 'AP' values across all categories present in the dataset. Additionally, 'mAP 0.5' signifies the 'AP' value when Intersection over Union (IOU) is set at 0.5. To validate the efficacy of both YOLOv5 and our enhanced target detection algorithm, we conducted tests using the Kitti dataset. The resulting Precision-Recall (P-R) curves are visually portrayed in Figure 10.

A comparison of the data depicted in the figure reveals a distinct trend: the enhanced algorithm consistently demonstrates significantly superior detection results when compared to the YOLOv5 algorithm on the KITTI dataset. Notably, the improved algorithm showcases a substantial enhancement in 'mAP 0.5,' escalating from 92.4% to 94.3%, reflecting a noteworthy advancement of 1.9 percentage points. This progress is especially prominent at the category level: the 'AP' for the vehicle category witnesses a 0.4 percentage point increase, rising from 97.8% to 98.2%. Similarly, the 'AP' for the pedestrian category exhibits a substantial 2.9 percentage point improvement, up from 87.1% to 90.4%. This discernible elevation in detection accuracy applies to both vehicle and pedestrian categories, highlighting the algorithm's efficacy in rectifying the oversight of small pedestrian targets within complex road scenarios. Moreover, the proposed algorithm excels in robustly identifying vehicles and pedestrians even amidst occlusion scenarios.

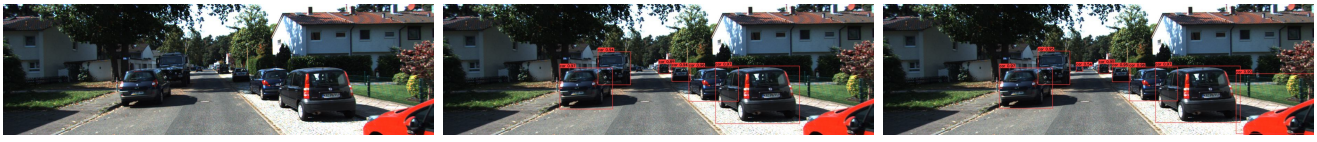
TABLE I: comparative experiment of attentional mechanisms

Method	Parameters	mAP 0.5
SPPF	7.01M	92.4
SPPF+CBAM	7.05M	92.4
SPPF+SE	7.04M	92.5
SPPF+CA	7.04M	92.7

To substantiate the enhancement in model detection stemming from the incorporation of a coordinate attention mechanism subsequent to SPPF, we conducted a series of attention comparison experiments. Specifically, we introduced three widely adopted attention mechanisms—CA, SE and CBAM—into the SPPF structure. The structure of the remaining components was retained across all variants. A thorough comparison was then undertaken on the KITTI dataset. The ensuing experimental outcomes are comprehensively presented in Table I.

To further substantiate the efficacy and superiority of the algorithm presented in this paper, we conducted an experimental comparison between the prevailing mainstream target detection algorithms and our proposed algorithm using the KITTI dataset, as shown in Table II.

Upon comparing the experimental outcomes of distinct algorithmic models detailed in Table II, a conspicuous pattern emerges. The enhanced YOLOv5 model emerges as the pacesetter in terms of detection accuracy, surpassing the performance of contemporary mainstream algorithms. Notably, YOLO-X manages to achieve a detection accuracy comparable to ours, yet its substantial size impedes real-time detection, rendering it impractical for certain applications. Meanwhile, YOLOv4-tiny, heralded for its lightweight design, exhibits the swiftest detection speed. However, its detection accuracy, standing at a mere 64.2%, renders it



(a) Original image (b) YOLOv5 (c) Improved YOLOv5

(a) Original image (b) YOLOv5 (c) Improved YOLOv5

Fig. 11: Visualization results of the KITTI dataset.

TABLE II: Comparison of different detection algorithms based on KITTI dataset

Algorithm	Model Size/MB	P/%	R/%	mAP 0.5/%	FPS
Faster R-CNN	107.8	64.5	84.5	83.4	21
SSD	100.6	89.2	63.0	70.8	56
YOLOv4	244.9	90.4	78.3	86.5	33
YOLOv4-Tiny	24.1	78.1	54.8	64.2	103
YOLOX-X	378.3	92.6	87.4	93.9	25
YOLOv5	13.6	93.2	86.8	92.4	79
Improved-YOLOv5	14.8	93.5	87.7	94.3	67

less applicable to intricate road scenarios. To sum up, the algorithm advocated in this paper adeptly balances heightened detection accuracy with its lightweight design, delivering robust overall performance. The inference speed remains commendably satisfactory, successfully fulfilling the pragmatic requirements of vehicle and pedestrian detection tasks. To offer a more visually intuitive demonstration of the algorithm's superiority, Fig. 11 juxtaposes the detection outcomes of the enhanced model pre- and post-improvement on the KITTI dataset.

Fig. 11(a) depicts the original image, while Fig. 11(b) presents the detection outcomes obtained from the unmodified YOLOv5 algorithm. In contrast, Fig. 11(c) showcases the detection results derived from our enhanced model. A discernible comparison between the figures yields valuable insights. The YOLOv5 algorithm detected six vehicles, each with a confidence level exceeding 0.5. However, this scenario exposes potential instances of missed detection within the YOLOv5 results. In stark contrast, our improved algorithm successfully identified a total of eight vehicles with confidence levels surpassing 0.5. This encompassed both distant small target vehicles and heavily obscured vehicles situated in the lower right. For the small target pedestrian situated adjacent to the tree on the left-hand side of the figure, the YOLOv5 algorithm registered a detection confidence of 0.90. Remarkably, our algorithm attained an elevated detection confidence of 0.97 for the same pedestrian. This disparity highlights our algorithm's pronounced proficiency in detecting diminutive targets and targets subject to occlusions. Overall, our algorithm demonstrates remarkable efficacy in detecting KITTI dataset scenarios, outperforming the original YOLOv5 algorithm and concurrently exhibiting substantial reduction in the missed detection rate.

C. Analysis of ablation experiment

This paper introduces three distinct improvement strategies. To validate the efficacy of each proposed module and to dissect the influence of each module on the YOLOv5 algorithm, we have designed an ablation experiment section. Our approach involves a stepwise integration of individual modules into the original YOLOv5 algorithm. This process permits a systematic evaluation of the effects arising from various module combinations on the final model's detection performance. The outcomes of these ablation experiments are presented in Table III. Here, the numerical labels assigned to each module serve to distinguish their respective contributions: ① signifies the enhanced multi-scale feature fusion module, ② designates the coordinate attention mechanism module, and ③ represents the SIOU optimized loss function module.

TABLE III: Ablation experiments

Experiment	Exp1	Exp2	Exp3	Exp4	Exp5	Exp6	Exp7
①	✓				✓	✓	✓
②		✓		✓		✓	✓
③			✓	✓	✓		✓
mAP 0.5	93.2	92.6	92.7	93.4	93.7	94.0	94.3

As evident from the table, the incorporation of each module detailed in this paper has yielded varying degrees of improvement in the algorithm's detection accuracy. Particularly noteworthy is the pronounced advancement attributed to the multiscale feature fusion module, which contributes to a 0.8% enhancement in the model's performance. This enhancement underscores the module's role in augmenting the model's capability for effective feature fusion. Furthermore, the introduction of the attention mechanism and the optimized loss function has resulted in discernible enhancements in the algorithm's detection performance. This substantiates the efficacy of the proposed algorithm in effectively addressing the intricate challenges of vehicle and pedestrian detection in complex road scenarios.

VI. Conclusion

This study introduces an enhanced vehicle and pedestrian target detection algorithm built upon a modified YOLOv5 architecture, and effectively applies it to real-world scenarios. Our model exhibits a significant improvement, boosting detection accuracy by 1.9 percentage points in comparison to the original YOLOv5 algorithm. In contrast to alternative algorithms, our refined approach achieves higher detection accuracy while preserving the advantage of algorithmic speed. This performance augmentation is particularly evident in our model's prowess in detecting small and obscured targets within genuine road conditions. Furthermore, our approach lays the groundwork for practical vehicle and pedestrian detection on hardware platforms, establishing itself as an efficient and effective target detection model. Looking ahead, our research trajectory will focus on optimizing the model's weight without compromising detection accuracy.

References

[1] P. Seuwou, E. Banissi, and G. Ubakanma, "The future of mobility with connected and autonomous vehicles in smart cities," *Digital Twin Technologies and Smart Cities*, pp. 37–52, 2020.

[2] S. Vimal, B. Ajay, and P. Thiruvikiraman, "Context pruned histogram of oriented gradients for pedestrian detection," in *2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*, pp. 718–722, 2013.

[3] J. Zhuang, "Compressive tracking based on hog and extended haar-like feature," in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pp. 326–331, 2016.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "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, 2014.

[5] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, 2015.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, 2016.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "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, 2016.

[9] J. Cao, Y. Pang, and X. Li, "Exploring multi-branch and high-level semantic networks for improving pedestrian detection," *arXiv preprint arXiv:1804.00872*, 2018.

[10] H. Nguyen, "Improving faster R-CNN framework for fast vehicle detection," *Mathematical Problems in Engineering*, vol. 2019, pp. 1–11, 2019.

[11] Z. Chen, H. Guo, J. Yang, H. Jiao, Z. Feng, L. Chen, and T. Gao, "Fast vehicle detection algorithm in traffic scene based on improved SSD," *Measurement*, vol. 201, pp. 263–270, 2022.

[12] G. Xiaoying, L. Qiaoling, Q. Zhikang, and X. Yan, "Target detection of forward vehicle based on improved SSD," in *2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, pp. 466–468, 2021.

[13] D. Yuan and Y. Xu, "Lightweight vehicle detection algorithm based on improved yolov4." *Engineering Letters*, vol. 29, no. 4, pp. 1544–1551, 2021.

[14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, 2017.

[15] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, 2018.

[16] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10 781–10 790, 2020.

[17] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, 2018.

[18] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

[19] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13 713–13 722, 2021.

[20] Z. Gevorgyan, "SIoU loss: More powerful learning for bounding box regression," *arXiv preprint arXiv:2205.12740*, 2022.