SAM-YOLO: An improved small object detection model for vehicle detection

JiaWang Liao [a], SuYu Jiang [a], MingHua Chen [a] and ChengJiao Sun [a,*]

[a] *School of Computer Engineering, Hubei University of Arts and Science, 296 Longzhong Road, Xiangyang, 441053, Hubei, China*
E-mails: sakurapuare@hbuas.edu.cn, jiao1952@126.com

Abstract. Vehicle detection using computer vision plays a crucial role in accurately recognizing and responding to various road conditions, targets, and signals, particularly within autonomous driving technology. However, traditional vehicle detection algorithms suffer from slow detection speed, low accuracy, and poor robustness. To address these challenges, this paper proposes the SAM-YOLO algorithm. SAM-YOLO incorporates the SimAM attention mechanism into the YOLOv7 network, allowing for the capture of more detailed information without introducing additional parameters. In this study, we experimentally redesigned the backbone network of SAM-YOLO by replacing the redundant part of the network layer with the C3 module, resulting in improved model performance while maintaining accuracy. The experimental results show that the SAM-YOLO algorithm performs excellently in several evaluation metrics under conventional conditions, especially outperforming other algorithms in accuracy and mAP values. In tests on the ExLight dataset facing extreme lighting conditions, SAM-YOLO similarly demonstrated optimal detection capabilities, especially in terms of robustness when dealing with complex lighting variations. These findings emphasize the potential of the SAM-YOLO algorithm for real-time and accurate target detection tasks, especially in environments with highly variable lighting conditions.

Keywords: Vehicle detection, SimAM attention mechanism, Small object detection

1. Introduction

In recent years, substantial advancements in autonomous driving technology have occurred, motivated by the pursuit of scientific and technological innovation, as well as the increasing demand for convenient travel [1, 2]. Autonomous driving technology empowers vehicles to perceive and comprehend their surroundings, formulate navigation plans, and regulate their movements without human intervention [3]. To accomplish this, a car must possess the capability to detect objects in its vicinity, discern road conditions, and make informed decisions regarding its trajectory [4]. Hence, achieving precise detection and recognition of vehicles and road environments is crucial for fully exploiting the capabilities of autonomous driving technology [5]. In this context, the development of machine learning models for vehicle visual detection has emerged as a crucial research area with substantial practical implications [6].

Traditional machine learning algorithms commonly used for object detection rely on manual feature engineering, including predefined feature extraction [7–9], sliding windows [10–12], and statistical learning [13–16]. These algorithms extract features from input images and utilize machine learning techniques to ascertain the presence of objects at each location [17]. The final detection outcome is obtained by aggregating multiple detection results using suppression rules. However, these algorithms face limitations when dealing with complex scenes [18], primarily due to the diverse shapes and viewpoints of detected objects, resulting in high computational complexity, low accuracy, and poor robustness [19]. Various factors like different driving poses, changes in lighting conditions, occlusion by surrounding objects, and interference from cluttered backgrounds pose challenges to traditional machine learning

[*] Corresponding author. E-mail: jiao1952@126.com.

object detection algorithms. The advent of deep learning has attracted significant attention in the field of artificial intelligence, particularly in the development and application of deep learning-based object detection algorithms [19].

YOLO (You Only Look Once) is an object detection algorithm based on convolution neural networks that was proposed by [20]. In contrast to two-stage object detection methods [21–24], YOLO can precisely predict the bounding box and object probabilities of the entire image in a single evaluation using a single neural network. This property makes YOLO an efficient approach for object detection since the entire detection process is contained within a single neural network, featuring a single end-to-end architecture that encompasses all processing steps from image input to output. The high effectiveness and efficiency of YOLO have contributed to its popularity as an algorithm in the field of computer vision, where it has found applications in various areas including autonomous driving, surveillance, and robotics [25].

YOLOv7 is a part of the YOLO family of object detection models [26]. It represents an enhancement over YOLOv5 [27]. Like the YOLO algorithm, YOLOv7 employs a single neural network to conduct an overall prediction for the entire image within one evaluation. As a conventional convolution neural network model, YOLOv7 comprises four primary components: the Input network, Backbone network, Neck network, and Head network. These components collaborate harmoniously to efficiently and precisely identify objects in images, making YOLOv7 a versatile tool applicable to a broad range of computer vision tasks.

Despite exhibiting exceptional performance in object detection tasks, the YOLO algorithm has high rates of missed detections and false alarms for detecting small objects [28–30]. Researchers have proposed various methods to address this issue [31], such as multi-scale feature representation [32–36], additional detection heads [37, 38], image enhancement [39], super-resolution techniques, and attention mechanisms.

For instance, Hsu and Lin proposed a multi-scale feature representation that combines length and width information, alleviating image distortion after resizing and integrating complementary data from multiple sub-images [40]. Carrasco et al. integrated features extracted from local images at different scales into the YOLOv5 Backbone network, effectively reducing the number of trainable parameters and floating-point operations per second [41]. As a result, both inference speed and accuracy were improved.

Zhu et al. presented a multi-sensor multi-level improved convolution network model that incorporates an improved reasoning head and feature fusion method, integrating radar data [42]. Additionally, Zhao et al. introduced a prediction head to YOLOv7 and utilized the SimAM module to enhance the detection of small objects or individuals [43].

Enhancing image information is also a prevalent approach in recent studies. Liu et al. employed the Flip-Mosaic algorithm to enhance the network's capability in detecting small targets and mitigating the false detection rate of occluded vehicle targets [44]. Likewise, Jiang et al. incorporated the attention mechanism and merged the infrared image with the image enhancement algorithm and the Global Attention Mechanism (GAM), resulting in enhanced accuracy for small target detection Jiang et al.. The method proposed by Shen et al. is based on multiple information perception and attention modules, including five processes: information preprocessing, information collection, information interaction, feature fusion, and attention generation [46].

Thus, this paper proposes an improved YOLOv7 object detection algorithm called SAM-YOLO that improves the accuracy of object positioning and recognition, while preserving the original excellent features of the YOLOv7 network. The contributions of this paper on-road road-vehicle visual detection can be summarized as follows:

- The SAM-YOLO algorithm introduces a Simple Attention Mechanism (SimAM) that integrates both channel-level and spatial-level information to model multidimensional dependencies, structural information, and global insights. This attention mechanism focuses selectively on critical regions within the image, thereby enhancing the precision of small object detection. By effectively capturing essential features within the two-dimensional space of images, it addresses information loss and improves the accuracy of behavior recognition.
- The SAM-YOLO algorithm's network layer has been reduced based on the concept of model lightweight design. This reduction significantly alleviates the computational burden caused by the multi-layer propagation of information during the inference process, thereby enhancing recognition speed and achieving high computational efficiency. Consequently, it becomes well-suited for fast image processing and rendering, making it suitable for real-time applications. Moreover, it facilitates the deployment of the algorithm on low-power vehicle terminals.

- We propose a new application of the SAM-YOLO algorithm specifically for detecting moving vehicles on the road. Our findings demonstrate that the SAM-YOLO algorithm offers advantages in performance when compared to existing YOLO and other algorithms.

The remaining sections of this paper are organized as follows. In Section 2, a restatement of the problem is provided, followed by the details of the SAM-YOLO algorithm in Section 3. The experimental results and effectiveness evaluation of the approach are presented in Section 4. Finally, the findings are summarized, and potential avenues for future research are discussed in Section 5.

2. Restatement of the problem

Object detection, or object recognition, is a fundamental problem in computer vision that involves identifying and localizing objects within images or video sequences. The task requires the model to predict both the presence and category of objects, as well as draw bounding boxes around detected objects to indicate their locations. This problem combines elements of classification and localization and poses challenges due to variations in object appearance, scale, occlusion, and environmental conditions.

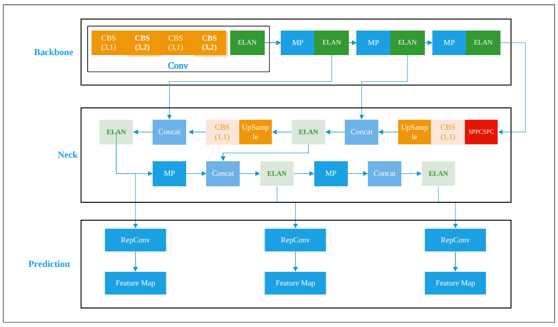


Figure 1. The architecture of YOLOv7

The YOLOv7 algorithm is widely applied in diverse object detection scenarios, and its network model is predominantly composed of Input, Backbone, Neck, and Head components, as shown in Figure 1. More specifically, the input layer consists of preprocessed and normalized image inputs, and the backbone network is responsible for extracting features from input images. The Head layer in YOLOv7 is a CSPSPP layer, so it is merged into the Neck layer in the image.

The multi-layer ELAN (Efficient Layer Aggregation Networks) structure is designed to enhance computational efficiency and strengthen feature fusion capabilities. This structure employs complex layer aggregation strategies to significantly improve feature extraction performance, making it more suitable for object detection tasks. Specifically, the ELAN architecture consists of basic units made up of multiple convolutional layers, activation functions (such as ReLU or Leaky ReLU), and normalization layers (such as Batch Normalization), which are further combined into higher-level groups. ELAN achieves parallel feature processing by aggregating outputs from multiple convolution paths at specific nodes, thereby enriching feature expression diversity. At the same time, ELAN emphasizes multi-scale feature aggregation, integrating information from different layers through feature fusion, and it alleviates the vanishing gradient problem with skip connections similar to those in ResNet, thus enhancing training stability and efficiency.

However, the original multi-layer ELAN structure in YOLOv7 results in substantial inter-layer information exchange, thereby decelerating the algorithm's training speed. Additionally, the utilization of fixed anchor sizes in YOLOv7 confines its effectiveness in discerning and detecting objects with various scales, particularly in demanding scenarios. These demanding scenarios can include conditions such as poor lighting, occlusion, or objects in high-speed motion, which complicate the detection process.

In object detection, another major challenge is the limitation of the YOLO algorithm in detecting small targets, especially in complex traffic environments. Due to the small size of these targets and their low pixel resolution, they often contain limited information in the image, making them susceptible to interference or occlusion from the background.

It is widely recognized that increasing the number of parameters and utilizing more intricate networks can partially enhance the accuracy of algorithmic detection. However, concerning detection accuracy, the effectiveness of improving training time and model size is restricted. Moreover, in engineering applications, the use of complex networks and a high volume of parameters is non-ideal owing to the computational constraints at the application level. Therefore, the presented algorithm strives to enhance the efficiency of the network layers instead of further augmenting the complexity of the YOLOv7 base model.

To tackle the challenges of detecting small targets in complex scenes, this paper proposes an improved YOLOv7 object detection algorithm based on the YOLOv7 network. The algorithm enhances the accuracy of target localization and recognition while retaining the fundamental features of the YOLOv7 network. This model differs from existing methods in that it does not require multi-scale feature fusion. Instead, it introduces the SimAM attention mechanism, which enables the network to learn and emphasize important aspects of the targets without introducing additional parameters. Additionally, the algorithm improves efficiency while maintaining detection accuracy by redesigning the original backbone network. By replacing the loss function used in the YOLOv7 algorithm, its recognition capability is enhanced, improving both parallelism and stability. Experimental results demonstrate that the improved YOLOv7 algorithm performs exceptionally well in handling complex scenes and small targets, effectively overcoming the aforementioned challenges.

3. SAM-YOLO algorithm

SAM-YOLO is an improved target detector based on the YOLOv7 architecture. The algorithm focuses on the challenges of small target detection in complex scenarios and improves the accuracy of target localization and identification while retaining the basic features of the YOLOv7 network while minimizing the potential degradation of target detection accuracy and recall. In addition, SAM-YOLO effectively reduces the number of parameters required for the model and speeds up the inference of the model. The network structure is shown in Fig. 3, and the main improvements are as follows:

(1) Incorporating the SimAM mechanism into the backbone network by designing experiments
(2) Redesigning the backbone network of the model
(3) Replacing part of the original structure with the more lightweight C3 module
(4) Redesigning the loss function of the model

3.1. SimAM attention mechanism

The attention mechanism is a widely used technique in the fields of machine learning and deep learning. It aims to simulate the human attention mechanism, selectively focusing on important parts of the input data. The attention mechanism has been extensively studied and applied in various tasks, including natural language processing, computer vision, and speech recognition. By introducing the attention mechanism, models can pay more attention to the parts that are more important for the current task when processing large amounts of information. The core idea of this mechanism is to determine the importance of each element in the input data through learning weight allocation. In the attention mechanism, each element can be assigned a weight or attention score, which reflects the model's degree of attention to each element. The model can adaptively adjust these weights based on the characteristics of different tasks and input data, thereby making more accurate predictions and processing.

The application of attention mechanisms in the fields of machine learning and deep learning is very extensive, including several important attention mechanisms such as CBAM, CA, SE, and SimAM. [47–51]

CBAM (Convolutional Block Attention Module) is an attention mechanism based on convolutional neural networks. It enhances model performance by capturing both channel attention and spatial attention. Channel attention is used to determine the importance of each channel in the input feature map, thereby weighting the channels. Spatial attention, on the other hand, determines the importance of each spatial position in the feature map, thus weighting the elements at different spatial positions. By combining channel attention and spatial attention, the CBAM attention mechanism enables the model to more accurately focus on the important parts of the input data.

The CA (Channel Attention) mechanism focuses on channel attention to determine the importance of each channel in the feature map. By utilizing global average pooling and fully connected layers, the CA attention model can compute and allocate weights for each channel to better capture the feature representations of different channels. The CA attention mechanism performs well in computationally intensive tasks and helps the model differentiate the importance of each channel more effectively.

The SE (Squeeze-and-Excitation) attention mechanism is a lightweight attention model that enhances the representation capability of the model efficiently. The core idea of the SE attention model is to dynamically adjust the weights of each channel by utilizing global contextual information. By introducing the "squeeze" and "excitation" stages, the SE attention mechanism can adaptively learn the importance of each channel and re-weight the features accordingly. The SE attention mechanism has achieved good results in many image classification and object detection tasks.

Yang et al. propose a module that efficiently generates true 3D weights in SimAM [52]. Specifically, it estimates the importance of individual neurons by taking into account the phenomenon in neurology where over-excited neurons usually inhibit surrounding neurons. This phenomenon suggests that neurons with spatial inhibition effects should be assigned higher weights in visual processing. The importance of each neuron is determined based on its linear separation from other neurons using the formula defined as Equation 1.

$$e_t^* = \frac{4(\hat{\sigma}^2 + \lambda)}{(t - \hat{\mu})^2 + 2\hat{\sigma}^2 + 2\lambda} \tag{1}$$

Theoretically, $\hat{\mu}$ represents the average of individual neurons, and $\hat{\sigma}$ denotes the variance of individual neurons. Equation 1 demonstrates that a lower neuron energy e^t indicates a greater dissimilarity between the neuron and its neighboring neurons, resulting in a correspondingly higher weight. λ is a hyperparameter introduced to stabilize the attention map during its computation. It represents a small positive value added to prevent division by zero and to improve numerical stability in the calculation of the attention weights. If λ is too small, the division by nearly zero might lead to extremely high attention values, which might dominate the learning process and lead to poor generalization. Conversely, if it's too large, it may overly smooth out the differences in attention, leading to underfitting. In our cases, $1e^{-4}$ was used in our experiment. Consequently, the significance of a neuron can be determined using $\frac{1}{e_t^*}$. Moreover, the algorithm reintroduces the concept of neurology and proposes that attention regulation in the mammalian brain commonly involves the amplification or scaling of neuronal responses, instead of mere addition or subtraction. Thus, the algorithm applies a scaling operation to the neuronal energy e^t to amplify its characteristics. Here, E represents the sequence of energy e^t of all neurons in both channel and space.

Computationally, $\hat{\mu}$ represents the average value of all neurons within a channel over the spatial dimensions (i.e., height H and width W).

$$\hat{\mu} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{:,:,i,j} \tag{2}$$

where $X_{:,:,i,j}$ represents all neurons at position (i, j) within the channel.

Variance $\hat{\sigma}^2$ describes the degree of dispersion of neurons around the calculated spatial mean. It is computed using the formula:

$$\hat{\sigma}^2 = \frac{1}{H \times W - 1} \sum_{i=1}^{H} \sum_{j=1}^{W} (X_{:,:,i,j} - \hat{\mu})^2 \tag{3}$$

Here, "-1" is used to provide an unbiased estimate.

Neuron energy e^t reflects the similarity of a neuron to its neighboring neurons. Lower energy indicates a greater difference between the neuron and its neighbors, thus possibly being more significant.

$$\tilde{X} = \frac{1}{e_t^*} \tag{4}$$

The importance of each neuron is determined by the linear separability defined in Equation 4, and the sigmoid function is employed to prevent the reciprocal of E from becoming excessively large. Moreover, as the *sigmoid* function is monotonically increasing, it preserves the relative weights between neurons.

Table 1

Test Result On Different Attention Mechanism

Mechanism	Params(%)	Precision %	Recall %	$mAP_{0.5}$	$mAP_{0.5:0.95}$
SimAM	+0%	0.958	0.889	0.895	0.684
CBAM	+0.5%	0.957	0.859	0.905	0.700
CA	+0.2%	0.961	0.852	0.905	0.695
SEAM	+0.9%	0.905	0.830	0.883	0.608

To determine the impact of the SimAM module on different parts of YOLOv7, we conducted a series of experiments to evaluate the architecture of the SimAM module that has the greatest positive impact on evaluation indicators. Specifically, we integrated the SimAM module into the input network, backbone network, neck network, and head network of YOLOv7 by replacing certain layers within the original architecture. We then compared the model performance before and after the SimAM module's integration. The experimental results show that the SimAM module has the greatest impact on the neck network of YOLOv7. After introducing the SimAM module, the performance of the backbone network of YOLOv7 has significantly improved. Table 1 shows detailed experimental results on our collected datasets.

Table 2

The Impact of SimAM Module on YOLOv7

Location	Precision %	Recall %	$mAP_{0.5}$	$mAP_{0.5:0.95}$
Backbone (Layer 12)	0.890	0.773	0.844	0.556
Neck (Layer 64)	0.928	0.820	0.886	0.607
Neck (Layer 76)	0.958	0.889	0.895	0.684
Neck (Layer 102)	0.922	0.863	0.895	0.642

In the SAM-YOLO model, the SimAM attention mechanism substitutes a segment of the ELAN structure within the Neck layer. More precisely, the SimAM module replaces the initial six convolutional layers, featuring 1, 1 as the step size parameter and 3, 1 as the convolution kernel size parameter. During forward propagation, SimAM evaluates the neurons and activates them based on Equation 4.

3.2. *Loss function*

In machine learning, the loss function serves as a metric for evaluating the discrepancy between the predicted and actual values of a model. By continuously adjusting its parameters to minimize this discrepancy, the model's performance can be improved, leading to better detection and prediction accuracy.

Meanwhile, during object detection, the algorithm generates multiple bounding boxes with high confidence around the target object. However, only one bounding box can accurately represent the target. To address this redundancy, a non-maximum suppression (NMS) algorithm is implemented, which ensures that only the most appropriate bounding box is selected. The algorithm starts by sorting all bounding boxes and then calculates the Intersection over Union (IoU) of the highest-confidence bounding box with the remaining boxes. If the IoU of a bounding box exceeds a predefined threshold, it is discarded.

To evaluate the performance of image segmentation models, the Intersection over Union (IoU) metric is commonly employed. This metric quantifies the degree of overlap between predicted and ground truth boxes, thereby assessing the accuracy of predictions made by the model.

$$IoU = \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|} \tag{5}$$

The formula for calculating IoU, as presented in Equation 5, typically utilizes the norm of IoU. In this equation, B represents the ground truth box with four parameters: x, y, w, and h, which respectively indicate the coordinates of the box center, as well as the width and height of the box. B^{gt} denotes the predicted box. The IoU-based loss function $\mathfrak{L}$ is defined in Equation 6. A smaller value of $\mathfrak{L}$ indicates a more effective detection outcome from the model.

$$\mathfrak{L}_{IoU} = 1 - \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|} \tag{6}$$

$$\mathfrak{L}_{SIoU} = 1 - IoU + \frac{\Delta + \Omega}{2} \tag{7}$$

Building upon this foundation, the $CIoU$ loss function used in YOLOv7 is replaced with the $SIoU$ loss function proposed by Gevorgyan and Zhora [53]. This loss function comprehensively considers four aspects: angle, distance, shape, and IoU. The resulting loss function, represented in Equation 7, takes into account the contribution of Δ (the distance between the center points of the two boxes) and Ω (the difference in area between the two boxes), in addition to IoU. The schematic diagram of the $SIoU$ loss function is illustrated in Figure 2.

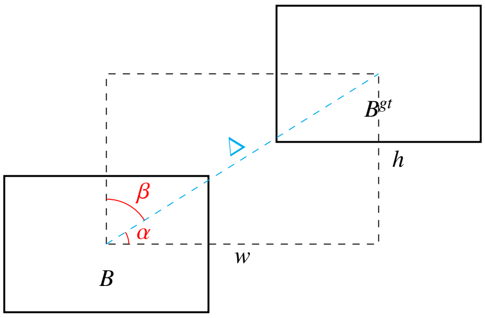


Figure 2. The schematic diagram of $SIoU$ loss function

Where Δ represents the distance between the center points of two boxes, Ω represents the difference in area between the two boxes, and IoU represents their intersection. A smaller value of $\mathfrak{L}$ indicates a better detection effect of the model.

The $SIoU$ loss function assigns different weights to object detection at various scales, giving more attention to objects with smaller scales during training. By introducing additional variables, such as shape loss, the $SIoU$ function not only provides a better measure of symmetry between the predicted box and the true box but also addresses the imbalance problem found in other IoU variants. Additionally, it facilitates faster convergence to the optimal solution and reduces training time. Moreover, it possesses greater sensitivity in detecting small target objects, thereby reflecting the effectiveness of the target detection model more accurately.

3.3. Construction network of the SAM-YOLO

In the SAM-YOLO algorithm, YOLOv7 is adopted as the basic network architecture, and the C3 module is incorporated. The number of layers and parameters in the network is reduced through this module, accelerating the model's inference and training speed.

Furthermore, the SimAM attention mechanism is introduced into the Neck network. This parameterless attention algorithm proposes an energy function based on mathematical methods to determine the importance of each neuron. Inspired by concepts in neurology, this approach avoids expending excessive energy on adjusting and enhancing the structure. Figure 3 illustrates the improved architecture of the algorithm.

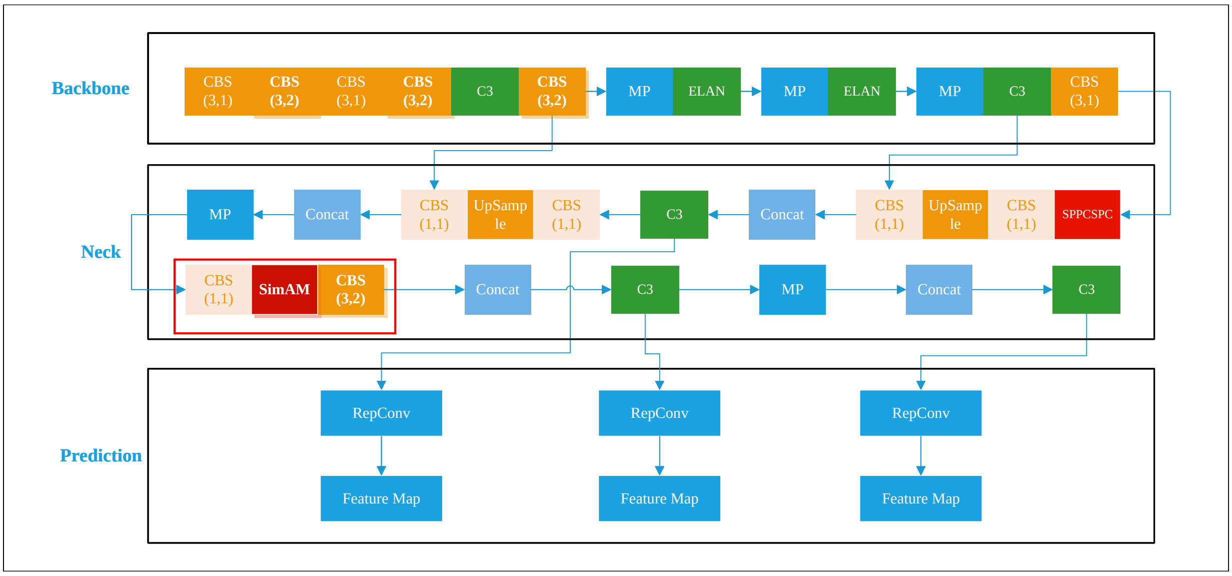


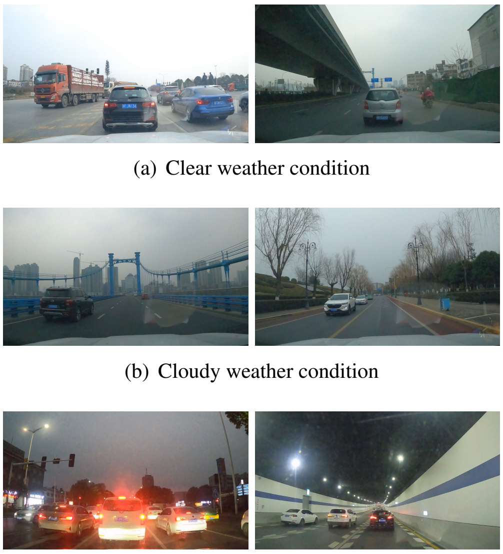
Figure 3. The architecture of the improved algorithm

4. Experiment and Analysis

4.1. The Dataset

To ensure the applicability of the vehicle recognition model on highways and urban roads, we collected a substantial number of authentic road videos captured by vehicle dashcams or built-in cameras near Xiangyang City, China, on highways and urban roads. The videos were captured from the driver's front-facing perspective encompassing diverse road and driving scenarios, including clear weather conditions on two-way four-lane roads, dimly lit rural road scenes, capturing traffic signs, vehicles, pedestrians, traffic lights, and road markings while vehicles are in motion. Figure 4(a), 4(b), 4(c) displays images from the training set, capturing different types of vehicles from various angles, road segments, and lighting conditions to enhance the dataset diversity. Figure 4(a) displays images captured in clear weather conditions, Figure 4(b) displays images captured in cloudy weather conditions, and Figure 4(c) displays images captured at night or in tunnels.

Segments with a high number of vehicles and clear video quality were meticulously selected, and one image sample was extracted for every 25 frames. Ultimately, a dataset of 16,008 images with vehicle information was obtained, encompassing various vehicle categories such as cars, trucks, taxis, and tankers. To ensure proper evaluation, the dataset was divided into training, validation, and test sets in a ratio of $7:2:1$. The training set comprises 11,206 frames, the validation set comprises 3,198 frames, and the test set comprises 1,604 frames.



(a) Clear weather condition

(b) Cloudy weather condition

(c) Night or tunnel condition

Figure 4. Images from the training set

4.2. Data Augmentation

Videos captured by dashcams or in-vehicle cameras, while partially representative of real road scenarios are influenced by various factors. Challenges such as image blurring from camera focus issues, underexposed or overexposed objects due to high contrast, and video noise from lighting conditions introduce recognition noise to the captured images. These issues hinder the training effectiveness of machine learning models.

To address this, we implement data augmentation during model training to enhance model robustness. Our augmentation techniques include:

(1) Rotation: Images are randomly rotated between -10° and $+10^\circ$ to help the model recognize objects regardless of their orientation.
(2) Shear: Horizontal and vertical shearing by $\pm 10^\circ$ simulates changes in perspective.
(3) Brightness Adjustment: Variation in image illumination by -30% to $+30\%$ enhances adaptation to fluctuating lighting conditions.
(4) Blur: A blur effect of up to 2 pixels approximates the out-of-focus images.
(5) Noise Addition: Introducing noise to up to 1% of pixels mimics sensor noise or transmission interference.

Additional challenges include motion-induced blur from the movement of objects and the camera's distance, often resulting in out-of-focus images. Gaussian blur, a linear smoothing filter, is utilized to reduce this blur while preserving edge information. The Gaussian kernel is defined as follows:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x - \frac{N-1}{2})^2 + (y - \frac{N-1}{2})^2}{2\sigma^2} \right) \tag{8}$$

Here, $G(x, y)$ denotes the weight at coordinates (x, y) within the 2D Gaussian kernel, which is normalized to ensure the total weight sum is 1.

Additionally, low-light conditions can introduce random image noise, which mere camera adjustments cannot correct. In this context, Poisson noisea statistical distribution that models random events like photon countingeffectively simulates noise under low light. This approach helps to replicate realistic imaging conditions, further improving the model's robustness.

4.3. Experimental environment

The experimental environment utilized in this study is summarized in Table 3.

Table 3

Experimental environment

Parameters	Value
Operating system	Ubuntu 23.10
CPU	Intel Xeon Platinum 8352V
GPU	RTX 4090 (24GB) * 32
RAM	1024GB
CUDA Version	12.1
Python Version	3.10.9
Framework	Pytorch
Pytorch Version	2.1

The training parameters utilized in this experiment are presented in Table 4.

Table 4

Training parameters

Parameters	Value
batch-size	64
epochs	300
img-size	640
hyp	hyp.scratch.p5.yaml
weights	yolov7.pt
workers	8

4.4. Evaluation Metrics

Performance evaluation of the SAM-YOLO algorithm involves the utilization of multiple metrics to assess the model's quality. For this study, the evaluation metrics employed include Precision Rate (P), Recall Rate (R), mean Average Precision with an IoU recognition threshold of 0.5 ($mAP_{0.5}$), and mean Average Precision within the IoU range of 0.5 to 0.95 ($mAP_{0.5:0.95}$).

Precision represents the probability of correctly predicting a positive sample, whereas recall denotes the probability of accurately identifying a positive sample from the original sample. $mAP_{0.5}$ refers to the mean average precision of the model with an IoU recognition threshold of 0.5, whereas $mAP_{0.5:0.95}$ represents the mean average precision with an IoU range of 0.5 to 0.95. The accuracies AP_i and mAP for each recognition threshold are computed according to Equation 9.

In Equation 9 and Equation 10, which establish the function $p(r)$ to represent the precision and recall rate, where the calculation equations of precision and recall are shown in Equation 11 and Equation 12. n in Equation 10 represents the total number of all samples. The area under the $p(r)$ curve in the interval $[0,1]$ represents the AP value.

In Equation 11 and Equation 12, TP represents the number of true positives, FP represents the number of false positives, and FN represents the number of false negatives.

$$AP = \int_0^1 p(r)dr \tag{9}$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \tag{10}$$

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

4.5. Experimental results

The effectiveness of the proposed method was evaluated through extensive experiments conducted on a benchmark dataset in this study. The experimental results indicate that the proposed SAM-YOLO algorithm achieves higher accuracy and recall rates compared to the original YOLOv7 algorithm. Additionally, there is a 3% improvement in the accuracy of $mAP_{0.5}$ and $mAP_{0.5:0.95}$.

Table 5

Comparison of evaluation indicators results

Model category	Precision (%)	Recall (%)	$mAP_{0.5}$ (%)	$mAP_{0.5:0.95}$ (%)	GFLOPS	Parameters (Millions)	Time (ms)	FPS(/s)
YOLOv7	96.45	92.15	95.06	75.44	105.5	35.5	12.6	79.36
SAM-YOLO	**96.34**	**93.46**	**95.96**	**75.72**	103.3	**36.5**	**9.5**	105.

Table 6

Comparison of detection category results

Model category	$mAP_{0.5}$(%)								$mAP_{0.5:0.95}$(%)							
	Total	People	Bicycle	Tricycle	Car	Motorcycle	Bus	Truck	Total	People	Bicycle	Tricycle	Car	Motorcycle	Bus	Truck
YOLOv7	95.0	86.7	91.8	98.6	98.1	93.0	99.2	98.4	75.5	52.0	61.4	81.1	82.9	65.7	84.5	84.8
SAM-YOLO	**95.9**	**88.2**	**94.3**	**99.2**	**98.3**	**93.6**	**99.3**	**98.3**	**75.5**	**54.8**	**65.3**	**82.0**	**84.2**	**66.2**	**86.3**	**85.7**

It can be observed from Figure 5 that the SAM-YOLO algorithm exhibits characteristics of missed detection rate and false detection rate across all detection categories, and demonstrates high accuracy in detecting small targets.

4.6. Ablation Experiment

In the ablation experiments conducted, by selectively adding or removing the SimAM, SIoU, and C3 modules, we were able to gain a deeper understanding of the specific impact of these components on the overall performance of the model. The addition or removal of each module provided us with unique insights, which in turn allowed us to meticulously evaluate their respective values and roles. The results of the ablation experiments show that when the SimAM, SIoU, and C3 modules are enabled simultaneously, the model can achieve the highest precision (0.961%), recall (0.930%), $mAP_{0.5}$ (0.962), and $mAP_{0.5:0.95}$ (0.725) with a 9.2% reduction in parameters while maintaining a relatively high frame rate (476 FPS). This suggests that the combined use of these modules can significantly improve the performance of the model.

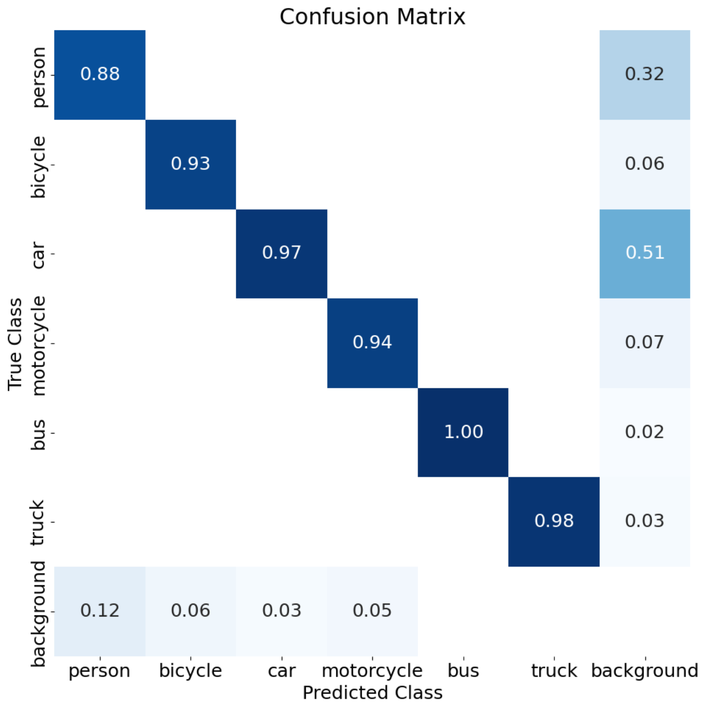


Figure 5. Confusion matrix of the SAM-YOLO algorithm

Table 7
Ablation experiment results

SimAM	SIoU	C3 Module	Params(%)	FPS	Precision %	Recall %	$mAP_{0.5}$	$mAP_{0.5:0.95}$
✓			0	476	0.958	0.889	0.895	0.684
	✓		0	312	0.946	0.853	0.908	0.649
		✓	-9.2%	312	0.944	0.860	0.913	0.658
✓	✓		0	434	0.933	0.847	0.925	0.623
✓		✓	-9.2%	434	0.958	0.862	0.905	0.667
	✓	✓	-9.2%	434	0.934	0.853	0.905	0.644
✓	✓	✓	-9.2%	476	0.961	0.930	0.962	0.725

4.7. Comparative Experiment

In this study, we compare and analyze the performance of multiple target detection algorithms in different contexts, including YOLOv7, SAM-YOLO, CBAM-YOLO, CA-YOLO, SEAM-YOLO, SSD, RetinaNet, and Faster-RCNN.We evaluate the performance of multiple target detection algorithms based on the number of parameters, Frames Per Second (FPS), precision, recall, and different mAP (mean Average Precision) metrics were comprehensively evaluated. All algorithms use ELAN or ResNet-50 as the underlying skeleton to ensure consistency and fairness in the evaluation.

In terms of overall performance, SAM-YOLO performs best in several metrics, especially in $mAP_{0.5}$ and $mAP_{0.5:0.95}$, which reach 0.962 and 0.725, respectively, and are significantly higher than other algorithms. This result indicates that SAM-YOLO is not only able to detect targets efficiently but also has high accuracy and robustness.YOLOv7 also shows strong performance, especially on the FPS metric, which reaches 476, which means that it can realize high-speed real-time target detection.

In the $mAP_{0.5}$ and $mAP_{0.5:0.95}$ evaluations for different object categories, SAM-YOLO has the best performance in almost all categories, especially in "tricycle", "bus" and "tricycle". Especially in the categories of "tricycle", "bus" and "truck", its $mAP_{0.5}$ is as high as 0.993, 0.987, and 0.991, respectively, which is much better than other algorithms. This further confirms the superiority of SAM-YOLO in dealing with complex scenes and targets of different scales.

In contrast, the performance of SSD, RetinaNet, and Faster-RCNN is relatively weak, especially in the $mAP_{0.5:0.95}$ metric, where these algorithms significantly lag behind the improved YOLO-based algorithms. In addition, the large parameter reduction of these algorithms, although conducive to reducing the computational complexity, may also affect the accuracy and comprehensive performance of the detection.

The experimental results show that SAM-YOLO and YOLOv7 are the optimal choices for target detection tasks with high speed and high accuracy. They are not only able to accurately detect various types of targets in complex scenes but also maintain high processing speed to meet the demands of real-time applications. Future research can further explore the optimization and adaptation of these algorithms in specific application scenarios to achieve higher performance and better adaptability.

Table 8
Comparison of detection results

Algorithm	Backbone	Params(%)	FPS	Precision %	Recall %	$mAP_{0.5}$	$mAP_{0.5:0.95}$
YOLOv7	ELAN	0	476	0.953	0.854	0.910	0.657
SAM-YOLO	ELAN	-9.2%	476	0.961	0.930	0.962	0.725
CBAM-YOLO	ELAN	+0.5%	300	0.957	0.859	0.905	0.700
CA-YOLO	ELAN	+0.2%	336	0.961	0.852	0.905	0.695
SEAM-YOLO	ELAN	+0.9%	340	0.905	0.830	0.883	0.608
SSD	ResNet-50	-32.1%	102	0.872	0.820	0.773	0.503
RetinaNet	ResNet-50	-31.3%	180	0.833	0.647	0.725	0.423
Faster-RCNN	ResNet-50	+17.0%	247	0.808	0.652	0.715	0.427

Table 9
Comparison of detection results on $mAP_{0.5}$

Algorithm	all	person	bicycle	tricycle	car	motorcycle	bus	truck
YOLOv7	0.910	0.870	0.899	0.950	0.931	0.860	0.949	0.935
SAM-YOLO	0.962	0.876	0.956	0.993	0.982	0.952	0.987	0.991
CBAM-YOLO	0.905	0.729	0.887	0.979	0.934	0.886	0.950	0.970
CA-YOLO	0.905	0.726	0.888	0.979	0.934	0.889	0.948	0.969
SEAM-YOLO	0.883	0.721	0.841	0.927	0.934	0.845	0.962	0.953
SSD	0.773	0.535	0.741	0.867	0.862	0.702	0.797	0.908
RetinaNet	0.725	0.495	0.678	0.792	0.857	0.647	0.757	0.848
Faster-RCNN	0.715	0.465	0.678	0.802	0.862	0.605	0.734	0.859

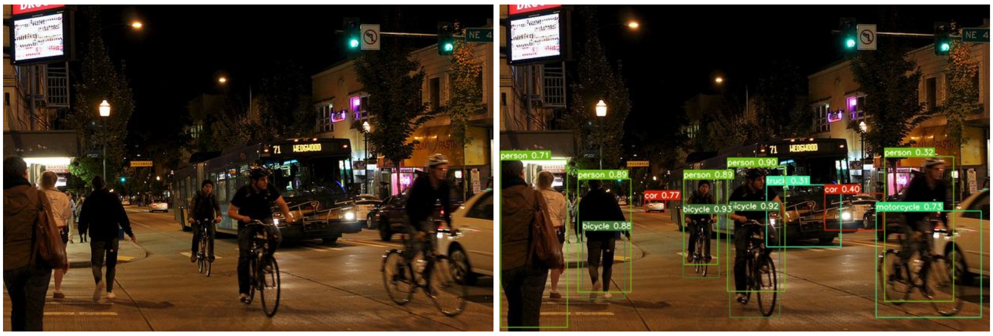
4.8. Test on ExDark Dataset

This section analyzes the target detection results on the ExDark dataset [54], which is designed to evaluate the performance of target detection algorithms under extreme lighting conditions. The analysis covers several algorithms including YOLOv7, SAM-YOLO, CBAM-YOLO, CA-YOLO, SEAM-YOLO, SSD, RetinaNet, and Faster-RCNN. By comparing the performance of each algorithm under two main metrics, $mAP_{0.5}$ and $mAP_{0.5:0.95}$, we aim to reveal the ability of different algorithms to adapt to extreme light change conditions.

Table 10

Comparison of detection results on $mAP_{0.5:0.95}$

Algorithm	all	person	bicycle	tricycle	car	motorcycle	bus	truck
YOLOv7	0.657	0.416	0.550	0.736	0.774	0.550	0.793	0.780
SAM-YOLO	0.725	0.494	0.792	0.678	0.857	0.647	0.757	0.848
CBAM-YOLO	0.700	0.469	0.637	0.772	0.86	0.564	0.768	0.829
CA-YOLO	0.695	0.443	0.614	0.793	0.791	0.609	0.797	0.82
SEAM-YOLO	0.608	0.362	0.51	0.683	0.739	0.500	0.726	0.738
SSD	0.503	0.327	0.413	0.365	0.788	0.345	0.596	0.684
RetinaNet	0.423	0.195	0.297	0.453	0.609	0.302	0.534	0.575
Faster-RCNN	0.427	0.192	0.295	0.481	0.613	0.282	0.537	0.585



(a) Original Image

(b) SAM-YOLO

In the $mAP_{0.5}$ metric, SAM-YOLO leads the other algorithms with a score of 0.593, which shows that it is more capable of detecting multiple types of objects in the ExDark dataset. Especially in the detection of "bicycle", "tricycle" and "truck", SAM-YOLO shows high accuracy. YOLOv7 also performs well, especially in the "car" category where it achieves a high score of 0.797. In contrast, CBAM-YOLO, CA-YOLO, and SEAM-YOLO perform slightly less well overall, with SEAM-YOLO in particular showing the weakest performance of all categories, probably due to its lack of adaptability to extreme light changes.

On the more stringent $mAP_{0.5:0.95}$ metric, SAM-YOLO leads again with a result of 0.382, further proving its better robustness and accuracy in dealing with complex lighting conditions. In addition, SAM-YOLO performs particularly well in the categories of "tricycle", "car" and "truck", which demonstrates its strong detection ability for large objects. This demonstrates its ability to detect large objects. In contrast, other algorithms such as CBAM-YOLO, CA-YOLO, and SEAM-YOLO perform poorly, especially CA-YOLO and SEAM-YOLO perform lower than expected on most of the categories, which may reflect their shortcomings in dealing with the capture of details under extreme lighting conditions.

By analyzing the test results on the ExDark dataset, we can conclude that the SAM-YOLO algorithm not only performs well under regular lighting conditions but also maintains a high detection performance under extreme lighting conditions. This ability makes it the preferred algorithm for high-precision target detection in complex lighting environments. However, the performance degradation of other algorithms under extreme lighting conditions suggests that improving the robustness of the algorithm to changes in lighting remains an important direction for future research.

5. Conclusion

In this paper, an improved YOLOv7 algorithm that incorporates a SimAM attention mechanism into the Neck network was proposed, replaces the $CIoU$ function of YOLOv7 with the $SIoU$ function, and simplifies the model

Table 11

Comparison of detection results on ExDark Dataset $mAP_{0.5}$

Algorithm	all	person	bicycle	tricycle	car	motorcycle	bus	truck
YOLOv7	0.521	0.318	0.415	0.455	0.797	0.342	0.622	0.701
SAM-YOLO	0.593	0.348	0.551	0.586	0.781	0.478	0.634	0.778
CBAM-YOLO	0.443	0.260	0.325	0.346	0.759	0.209	0.563	0.643
CA-YOLO	0.478	0.241	0.356	0.537	0.656	0.353	0.605	0.597
SEAM-YOLO	0.309	0.152	0.138	0.129	0.694	0.127	0.454	0.468
SSD	0.315	0.178	0.165	0.075	0.712	0.173	0.451	0.453
RetinaNet	0.314	0.118	0.159	0.310	0.545	0.136	0.456	0.471
Faster-RCNN	0.310	0.139	0.238	0.196	0.543	0.141	0.401	0.508

Table 12

Comparison of detection results on ExDark Dataset $mAP_{0.5:0.95}$

Algorithm	all	person	bicycle	tricycle	car	motorcycle	bus	truck
YOLOv7	0.282	0.111	0.153	0.231	0.522	0.119	0.428	0.408
SAM-YOLO	0.382	0.160	0.256	0.413	0.586	0.231	0.493	0.536
CBAM-YOLO	0.227	0.083	0.104	0.150	0.468	0.062	0.354	0.370
CA-YOLO	0.159	0.054	0.046	0.025	0.425	0.044	0.270	0.237
SEAM-YOLO	0.142	0.039	0.037	0.040	0.367	0.028	0.268	0.211
SSD	0.169	0.101	0.104	0.265	0.141	0.135	0.145	0.152
RetinaNet	0.136	0.101	0.090	0.170	0.010	0.100	0.124	0.104
Faster-RCNN	0.157	0.008	0.084	0.142	0.087	0.078	0.124	0.104

architecture to accelerate model training and reduce the number of parameters. This approach enhances the model's generalization ability, improves the learning of spatial features, and enhances computational efficiency. The results reveal that the SAM-YOLO algorithm outperforms other algorithms in terms of comprehensive performance, especially in terms of accuracy and mAP metrics, both in standard test environments and under extreme lighting conditions. This demonstrates the potential of SAM-YOLO in realizing high-speed and high-accuracy target detection, especially for real-world application scenarios with variable lighting conditions.

Regarding model training, the training set comprises real road videos with a fixed perspective effect, and the proportion of vehicles on the road varies. Consequently, the model training may result in bias. To mitigate this limitation, future studies should enhance the original dataset to alleviate the impact of issues related to data collection on the performance of the model. Furthermore, during the data collection process, non-conventional shooting perspectives like reverse and left-right angles should be introduced to enhance the capability of detecting vehicles in various angles and environments.

Additionally, our findings point to a general decrease in the performance of target detection algorithms under extreme lighting conditions, highlighting the importance of improving the robustness of algorithms to lighting changes. While SAM-YOLO performed the best in these tests, the performance degradation of the other algorithms hints at the need for future work, especially in optimizing the algorithms to better adapt to extreme lighting conditions.

Declarations

Funding

This work is supported by the Natural Science Foundation of Hubei Province (No. 2024AFB147), Xiangyang Science & Technology Plan (High-tech field, No. 2022ABH006596), Innovation and Entrepreneurship Education Special Project (No. CX2023003).

Conflict of interest

The author(s) declared no potential conflicts of interest concerning the research, authorship, and/or publication of this article.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] Grigorescu, S., Trasnea, B., Cocias, T., Macesanu, G.: A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics* **37**(3), 362–386 (2020)

[2] Kiran, B.R., Sobh, I., Talpaert, V., Mannion, P., Sallab, A.A.A., Yogamani, S., Perez, P.: Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems* **23**(6), 4909–4926 (2022)

[3] Yurtsever, E., Lambert, J., Carballo, A., Takeda, K.: A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access* **8**, 58443–58469 (2020)

[4] Petit, J., Shladover, S.E.: Potential cyberattacks on automated vehicles. *IEEE Transactions on Intelligent Transportation Systems* **16**(2), 1–11 (2014)

[5] Gupta, A., Anpalagan, A., Guan, L., Khwaja, A.S.: Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array* **10**, 10–57 (2021)

[6] Liu, L., Lu, S., Zhong, R., Wu, B., Yao, Y., Zhang, Q., Shi, W.: Computing systems for autonomous driving: State of the art and challenges. *IEEE Internet of Things Journal* **8**(8), 6469–6486 (2021)

[7] Outay, F., Mengash, H.A., Adnan, M.: Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: Recent advances and challenges. *Transportation Research Part A: Policy and Practice* **141**, 116–129 (2020)

[8] Wang, X., Liu, Y., Wang, F., Wang, J., Liu, L., Wang, J.: Feature extraction and dynamic identification of drivers' emotions. *Transportation Research Part F: Traffic Psychology and Behaviour* **62**, 175–191 (2019)

[9] Shi, X., Wong, Y.D., Li, M.Z.-F., Palanisamy, C., Chai, C.: A feature learning approach based on XGBoost for driving assessment and risk prediction. *Accident Analysis & Prevention* **129**, 170–179 (2019)

[10] Chen, X., Xiang, S., Liu, C.-L., Pan, C.-H.: Vehicle detection in satellite images by hybrid deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters* **11**(10), 1797–1801 (2014)

[11] Chen, Z., Huang, X.: Pedestrian detection for autonomous vehicle using multi-spectral cameras. *IEEE Transactions on Intelligent Vehicles* **4**(2), 211–219 (2019)

[12] Song, H., Liang, H., Li, H., Dai, Z., Yun, X.: Vision-based vehicle detection and counting system using deep learning in highway scenes. *European Transport Research Review* **11**(1), 1–16 (2019)

[13] Cucchiara, R., Grana, C., Piccardi, M., Prati, A.: Statistic and knowledge-based moving object detection in traffic scenes. In: *2000 IEEE Intelligent Transportation Systems. Proceedings*, pp. 27–32 (2000)

[14] Sun, Z., Bebis, G., Miller, R.: Monocular precrash vehicle detection: features and classifiers. *IEEE Transactions on Image Processing* **15**(7), 2019–2034 (2006)

[15] Wang, C.-C.R., Lien, J.-J.J.: Automatic vehicle detection using local features: A statistical approach. *IEEE Transactions on Intelligent Transportation Systems* **9**(1), 83–96 (2008)

[16] Alotibi, F., Abdelhakim, M.: Anomaly detection for cooperative adaptive cruise control in autonomous vehicles using statistical learning and kinematic model. *IEEE Transactions on Intelligent Transportation Systems* **22**(6), 3468–3478 (2021)

[17] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikainen, M.: Deep learning for generic object detection: A survey. *International Journal of Computer Vision* **128**(2), 261–318 (2019)

[18] Wang, K., Zhou, T., Li, X., Ren, F.: Performance and challenges of 3d object detection methods in complex scenes for autonomous driving. *IEEE Transactions on Intelligent Vehicles* **8**(2), 1699–1716 (2023)

[19] Srivastava, S., Narayan, S., Mittal, S.: A survey of deep learning techniques for vehicle detection from UAV images. *Journal of Systems Architecture* **117**, 102–152 (2021)

[20] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788 (2016)

[21] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440 (2015)

[22] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: Ssd: Single shot multibox detector. In: *Computer Vision – ECCV 2016*, pp. 21–37. Springer, Cham (2016)

[23] Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. *International Journal of Computer Vision* **128**(3), 642–656 (2019)

[24] Wang, R., Shivanna, R., Cheng, D., Jain, S., Lin, D., Hong, L., Chi, E.: DCN v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In: *Proceedings of the Web Conference 2021*, pp. 1785–1797 (2021)

[25] Li, G., Ji, Z., Qu, X., Zhou, R., Cao, D.: Cross-domain object detection for autonomous driving: A stepwise domain adaptative yolo approach. *IEEE Transactions on Intelligent Vehicles* **7**(3), 603–615 (2022)

[26] Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M.: Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv.org (2022)

[27] Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., Michael, K., TaoXie, Fang, J., imyhxy, Lorna, Yifu, Z., Wong, C., V, A., Montes, D., Wang, Z., Fati, C., Nadar, J., Laughing, UnglvKitDe, Sonck, V., tkianai, yxNONG, Skalski, P., Hogan, A., Nair, D., Strobel, M., Jain, M.: ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation. Zenodo (2022)

[28] Jiang, C., Ren, H., Ye, X., Zhu, J., Zeng, H., Nan, Y., Sun, M., Ren, X., Huo, H.: Object detection from UAV thermal infrared images and videos using YOLO models. *International Journal of Applied Earth Observation and Geoinformation* **112**, 102912 (2022)

[29] Li, R., Shen, Y.: YOLOSR-IST: A deep learning method for small target detection in infrared remote sensing images based on super-resolution and YOLO. *Signal Processing* **208**, 108962 (2023)

[30] Hu, J., Zhi, X., Shi, T., Zhang, W., Cui, Y., Zhao, S.: PAG-YOLO: A portable attention-guided YOLO network for small ship detection. *Remote Sensing* **13**(16), 3059 (2021)

[31] Liu, Y., Sun, P., Wergeles, N., Shang, Y.: A survey and performance evaluation of deep learning methods for small object detection. *Expert Systems with Applications* **172**, 114602 (2021)

[32] Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: *Lecture Notes in Computer Science*, vol. 9912. Cham, pp. 483–499 (2016)

[33] Hong, S., Roh, B., Kim, K.-H., Cheon, Y., Park, M.: Pvanet: Lightweight deep neural networks for real-time object detection. arXiv.org (2016)

[34] Najibi, M., Samangouei, P., Chellappa, R., Davis, L.S.: SSH: Single stage headless face detector. In: *2017 IEEE International Conference on Computer Vision*, pp. 4885–4894 (2017)

[35] Wu, X., Hong, D., Ghamisi, P., Li, W., Tao, R.: MsRi-CCF: Multi-scale and rotation-insensitive convolutional channel features for geospatial object detection. *Remote Sensing* **10**(12), 1990 (2018)

[36] Najibi, M., Singh, B., Davis, L.: Autofocus: Efficient multi-scale inference. In: *2019 IEEE/CVF International Conference on Computer Vision*, pp. 9745–9755 (2019)

[37] Zhu, X., Lyu, S., Wang, X., Zhao, Q.: Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios. In: *2021 IEEE/CVF International Conference on Computer Vision Workshops*, pp. 2778–2788 (2021)

[38] Deng, C., Wang, M., Liu, L., Liu, Y., Jiang, Y.: Extended feature pyramid network for small object detection. *IEEE Transactions on Multimedia* **24**, 1968–1979 (2022)

[39] Rabbi, J., Ray, N., Schubert, M., Chowdhury, S., Chao, D.: Small-object detection in remote sensing images with end-to-end edge-enhanced GAN and object detector network. *Remote Sensing* **12**(9), 1432 (2020)

[40] Hsu, W.-Y., Lin, W.-Y.: Adaptive fusion of multi-scale YOLO for pedestrian detection. *IEEE Access* **9**, 110063–110073 (2021)

[41] Carrasco, D.P., Rashwan, H.A., Garc'ıa, M., Puig, D.: T-YOLO: Tiny vehicle detection based on YOLO and multi-scale convolutional neural networks. *IEEE Access* **11**, 22430–22440 (2023)

[42] Zhu, J., Li, X., Jin, P., Xu, Q., Sun, Z., Song, X.: MME-YOLO: Multi-sensor multi-level enhanced YOLO for robust vehicle detection in traffic surveillance. *Sensors* **21**(1), 27 (2020)

[43] Zhao, H., Zhang, H., Zhao, Y.: Yolov7-sea: Object detection of maritime uav images based on improved yolov7. In: *2023 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pp. 233–238 (2023)

[44] Liu, S., Wang, Y., Yu, Q., Liu, H., Peng, Z.: CEAM-YOLOv7: Improved YOLOv7 based on channel expansion and attention mechanism for driver distraction behavior detection. *IEEE Access* **10**, 129116–129124 (2022)

[45] Jiang, K., Xie, T., Yan, R., Wen, X., Li, D., Jiang, H., Jiang, N., Feng, L., Duan, X., Wang, J.: An attention mechanism-improved YOLOv7 object detection algorithm for hemp duck count estimation. *Agriculture* **12**(10), 1659 (2022)

[46] Shen, X., Wang, H., Cui, T., Guo, Z., Fu, X.: Multiple information perception-based attention in YOLO for underwater object detection. The Visual Computer (2023)

[47] Cheng, A., Xiao, J., Li, Y., Sun, Y., Ren, Y., Liu, J.: Enhancing remote sensing object detection with k-cbst yolo: Integrating cbam and swin-transformer. *Remote Sensing* **16**(16), 2885 (2024)

[48] Shen, L., Lang, B., Song, Z.: Ca-yolo: Model optimization for remote sensing image object detection. Ieee Access (2023)

[49] Jia, L., Wang, T., Chen, Y., Zang, Y., Li, X., Shi, H., Gao, L.: Mobilenet-ca-yolo: An improved yolov7 based on the mobilenetv3 and attention mechanism for rice pests and diseases detection. *Agriculture* **13**(7), 1285 (2023)

[50] Wu, T., Dong, Y.: Yolo-se: Improved yolov8 for remote sensing object detection and recognition. *Applied Sciences* **13**(24), 12977 (2023)

[51] Mahaadevan, V., Narayanamoorthi, R., Gono, R., Moldrik, P.: Automatic identifier of socket for electrical vehicles using swin-transformer and simam attention mechanism-based evs yolo. IEEE Access (2023)

[52] Yang, L., Zhang, R.-Y., Li, L., Xie, X.: Simam: A simple, parameter-free attention module for convolutional neural networks. In: Meila, M., Zhang, T. (eds.) *Proceedings of the 38th International Conference on Machine Learning*, vol. 139, pp. 11863–11874 (2021)

[53] Gevorgyan, Z.: Siou loss: More powerful learning for bounding box regression. arXiv.org (2022)

[54] Loh, Y.P., Chan, C.S.: Getting to know low-light images with the exclusively dark dataset. *Computer Vision and Image Understanding* **178**, 30–42 (2019)

[55] Paneru, S., Jeelani, I.: Computer vision applications in construction: Current state, opportunities & challenges. *Automation in Construction* **132**, 103940 (2021)

[56] Wang, Z., Zhan, J., Duan, C., Guan, X., Lu, P., Yang, K.: A review of vehicle detection techniques for intelligent vehicles. IEEE Transactions on Neural Networks and Learning Systems, 1–21 (2022)

[57] Yang, Z., Pun-Cheng, L.S.C.: Vehicle detection in intelligent transportation systems and its applications under varying environments: A review. *Image and Vision Computing* **69**, 143–154 (2018)

[58] Zhang, Y., Guo, Z., Wu, J., Tian, Y., Tang, H., Guo, X.: Real-time vehicle detection based on improved YOLO v5. *Sustainability* **14**(19), 12274 (2022)

[59] Yan, J., Zeng, Y., Lin, J., Pei, Z., Fan, J., Fang, C., Cai, Y.: Enhanced object detection in pediatric bronchoscopy images using yolo-based algorithms with cbam attention mechanism. Heliyon **10**(12) (2024)