vehicles

MDPI

Article

MLRP-YOLOv8n: A Vehicle Target Detection Algorithm That Integrates Mixed Local Channel Attention and Large Kernel Separable Attention

Wenqiang Yu [1,*], Shui Yu [2], Qingmin Zhu [1] and Fangpeng Ning [3]

[1] School of Mechanical Engineering, Shandong Huayu University of Technology, Dezhou 253034, China; lionel_qm14@163.com
[2] School of Physical Science and Technology, Southwest Jiaotong University, Chengdu 610031, China; shuiyu@my.swjtu.edu.cn
[3] School of New Energy Vechicles and Engineering, Fuzhou Technology and Business University, Fuzhou 350715, China; strayxl@163.com
* Correspondence: ywq@huayu.edu.cn

Abstract

Autonomous driving, as a core component of intelligent transportation systems, relies highly on precise environmental perception capabilities. Vehicle target detection is the fundamental task of environmental perception. However, complex factors in real scenarios (such as target occlusion, illumination changes, and dense traffic flow) often lead to feature misjudgments, missed detections, target positioning deviations, and category confusions in existing methods. To address these challenges, this paper proposes the MLRP-YOLOv8n model that integrates Mixed Local Channel Attention (MLCA) and large kernel separable attention (LSKA). Three complementary attention mechanisms as well as improved regression loss are integrated into the lightweight YOLOv8n architecture to improve the accuracy of vehicle detection while maintaining computational efficiency. Firstly, MLCA is embedded in the C2f feature extraction module to enhance local feature focus; the SPPF module integrates LSKA optimize multi-scale feature fusion; RFCBAMConv convolution is used to replace the original convolution in the neck to enhance cross-level feature correlation; the PIoUv2 loss function is introduced instead of Complete Intersection over Union (CIoU) to accelerate model convergence and reduce regression errors. Experiments on the KITTI Detection dataset subset and UA-DETRAC datasets show that MLRP-YOLOv8n improves the mean average precision (mAP) by 1.9% and 3.2% respectively on the KITTI Detection dataset subset and UA-DETRAC datasets. This model achieves a balance between detection accuracy, tracking robustness, and computational efficiency, providing a reliable solution for autonomous driving environment perception.

Keywords: vehicle target detection; YOLOv8; MLCA; PIoUv2 attention mechanism

Check for updates

Academic Editors: Xubin Sun and Weifeng Zhong

Received: 3 April 2026
Revised: 30 May 2026
Accepted: 2 June 2026
Published: 4 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

With the rapid development of intelligent transportation, autonomous driving has become a key direction to solve urban traffic congestion and reduce accident rates [1]. Environmental perception, as the "visual system" of autonomous vehicles, directly determines the safety and reliability of the decision-making system. To achieve accurate multi-target positioning of vehicles, vehicle target detection is the core link [2]. However, real traffic scenarios pose severe challenges to the existing methods, as shown in Table 1.

Table 1. The current challenges.

Challenges	Content
Occlusion problem	Dense traffic flow leads to mutual occlusion between vehicles (occlusion rate can reach 40%), resulting in local feature loss and significantly increased missed detection rate.
Scale variation	Vehicles at different distances exhibit significant scale differences (from clear targets in close range to small targets in distant range), making it difficult for single scale models to adapt.
Dynamic interference	Changes in day and night lighting, camera vibrations (during vehicle operation) can cause unstable features, leading to trajectory drift or identity switching.

The development process of vehicle object detection technology can be divided into two stages: traditional artificial feature methods and deep learning driven. With the continuous improvement of hardware computing power and algorithm iteration, as well as the continuous enrichment of large-scale annotated datasets [3–5], deep learning-based methods have gradually become mainstream, giving rise to two major technical routes: two-stage object detection and single-stage object detection.

The traditional method is based on manually designing features and classifiers, and its process includes four key steps:

- Region selection: Generate candidate regions by traversing the image through a sliding window, or use a selective search algorithm [6] to merge superpixel regions based on underlying features such as color and texture. The former has an exponential increase in computational complexity with image size, while the latter reduces computational complexity through hierarchical grouping, but still requires the generation of approximately 2000 candidate boxes per image.
- Feature extraction: relies on manually designed feature descriptors, such as Histogram of Oriented Gradient (HOG) [7], to characterize the contour features of the target through statistical analysis of the gradient direction distribution in local areas of the image, but is sensitive to changes in lighting; Scale-Invariant Feature Transform (SIFT) [8] extracts key points and generates scale and rotation invariant feature descriptors, but it has high computational complexity.
- Feature classification: Linear classifiers such as Support Vector Machine (SVM) [9] are used to classify feature vectors. This type of classifier relies on linear or shallow nonlinear decision boundaries, making it difficult to model the diverse apparent features of vehicles in complex scenes.
- Post processing: Non-Maximum Suppression (NMS) [10] is used to remove redundant detection boxes, but it can easily lead to missed detection of small targets.

Although traditional methods have played an important role in early object detection tasks, they have significant limitations in terms of feature expression ability, environmental adaptability, and computational efficiency. For example, manually designed features are difficult to adapt to changes in lighting, occlusion, and target deformation in complex environments, and classifiers have difficulty modeling high-dimensional nonlinear features of targets, which limits their detection accuracy and generalization ability. Therefore, with the rise in deep learning technology, researchers have begun to explore neural network-based object detection methods.

The object detection method based on deep learning breaks through the dependence of traditional methods on manually designed features, directly learns feature representations from data, and gradually evolves into two major technical routes: two-stage detection and single-stage detection.

(1) Two-stage object detection algorithm

The core idea of the two-stage object detection algorithm is to generate region proposals on the input image that may contain the target, then extract features from the candidate regions and perform object recognition through fully connected networks or lightweight classifiers, while regression optimizing bounding boxes to improve detection accuracy. In the early stages of applying deep learning to object detection, researchers attempted to use convolutional neural networks for direct object localization. MultiBox [11] uses the CNN to directly generate category independent candidate boxes (anchor boxes), and jointly optimizes the bounding box position and confidence through the MultiBox loss function, laying the foundation for subsequent single-stage detectors. The R-CNN framework (Regions with Convolutional Neural Networks) proposed by Girshick et al. [12] demonstrated for the first time the effectiveness of supervised pretraining and domain fine-tuning strategies in object detection tasks by combining selective search with convolutional neural networks. This significantly improved the detection accuracy of the VOC 2012 dataset and demonstrated that deep features based on CNNs have stronger expressive power compared to manual features such as the HOG. He et al. [13] proposed the SPPNet (Spatial Pyramid Pooling Network), which introduces spatial pyramid pooling to enable the network to directly process input images of any size and convert them into fixed length feature vectors, significantly improving the efficiency of object detection. R. Girshick improved the R-CNN algorithm and proposed the Fast R-CNN [14] algorithm, which introduces an RoI (Region of Interest) pooling layer to map irregularly sized RoI regions to fixed size feature vectors, solving the problem of fixed input size in fully connected layers and significantly improving detection efficiency. S. Ren et al. [15] proposed the Faster R-CNN algorithm based on the Fast R-CNN algorithm, which significantly improves efficiency by introducing the Region Proposal Network (RPN) instead of traditional selective search algorithms. Lin et al. [16] proposed the Feature Pyramid Network (FPN) algorithm, which combines the semantic information of high-level features with the spatial details of low-level features through top-down paths and horizontal connections to construct a feature pyramid with multi-scale representation capability. Compared with traditional multi-scale methods, the FPN significantly reduces computational costs and improves detection accuracy by sharing feature maps. He et al. [17] proposed a new object detection and instance segmentation algorithm Mask R-CNN based on Faster R-CNN. This algorithm replaces the RoI Pooling layer with RoI Align to solve the problem of region mismatch caused by quantization operations in RoI Pooling, thereby improving localization accuracy. In addition, the Mask R-CNN adds a segmentation branch since the object detection branch, which uses a Fully Convolutional Network (FCN) for pixel-level classification, enabling the model to simultaneously achieve object detection and instance segmentation tasks.

(2) Single-stage object detection algorithm

Although the two-stage object detection method has made significant progress in accuracy, its complex candidate region generation and feature extraction process result in slow inference speed, making it difficult to meet real-time requirements. In this context, researchers have proposed a single-stage object detection framework, which was originally designed to directly predict the target category and bounding box on the input image by removing the candidate region generation step, thereby significantly improving detection efficiency. As an early attempt, OverFeat [18] proposed a multi-scale sliding window detection framework based on the CNN, which predicts the bounding box coordinates of the target through regression methods, achieving end-to-end detection and classification. The You Only Look Once (YOLO) detection algorithm proposed by Joseph et al. [19] is the first one-stage detection method that treats object detection tasks as a single regression problem. The algorithm divides the image into fixed grids and predicts multiple bounding boxes and category probabilities in each grid, achieving fast end-to-end object detection. W. Liu et al. [20] proposed the Single

Shot MultiBox Detector (SSD) detection algorithm, which inherits the prior box mechanism of MultiBox and simultaneously predicts bounding boxes on multiple levels of feature maps, constructing a multi-scale object detection framework. SSD utilizes shallow feature maps to capture small targets and deep feature maps to detect large targets, which not only significantly improves detection efficiency but also adapts well to the detection needs of targets of different sizes. Fu et al. [21] improved the SSD algorithm and proposed the Deconvolutional Single Shot Detector (DSSD) algorithm. The DSSD introduces a Deconvolution Module based on the SSD, which gradually restores the spatial details of low-resolution feature maps by adding multiple deconvolution layers on shallow feature maps. Combined with additional prediction branches, it significantly improves the detection ability of small targets. In addition, the DSSD retains the multi-scale feature detection framework of the SSD, allowing it to still have high performance in detecting large targets. Redmon et al. [22] proposed the YOLOv2 detection algorithm, which introduced a new backbone network Darknet-19 and significantly improved the model's feature extraction capability. In addition, the algorithm comprehensively introduces Batch Normalization (BN) into the network [23], which not only improves the accuracy of the model, but also accelerates the convergence speed of the training process. Redmon et al. [24] proposed the YOLOv3 detection algorithm, which uses a deeper and more efficient backbone network Darknet-53. Its design combines residual connections and more convolutional layers, enhancing feature extraction capabilities while achieving a good balance between depth and computational efficiency. To overcome the drawbacks of anchor box detectors, Law et al. [25] proposed the CornerNet algorithm. CornerNet abandons the traditional anchor box mechanism and instead predicts the position of the target by detecting key points in the upper left and lower right corners of the target bounding box in the input image, combined with feature nesting relationships. Duan et al. [26] proposed the CenterNet algorithm, which is an object detection method based on keypoint detection. It only detects the center point of the target object in the input image and determines the position of the bounding box by combining the width and height information of the target. This method significantly simplifies the post-processing steps, achieves significant improvements in detection speed and accuracy, and further reduces the complexity of the algorithm. YOLOv4 [27] introduces various optimization strategies, including Mosaic data augmentation to enhance the diversity of training data, CSPDarknet53 network structure to reduce redundant gradients and improve feature extraction efficiency, and a multi-scale feature fusion architecture combining FPN and PANet [28] to enhance detection accuracy. As another important development version of the YOLO series, YOLOv5 introduces the Focus layer to improve feature extraction efficiency, adopts adaptive anchor box calculation and automatic learning rate adjustment strategies to optimize the model training process, and provides multiple model scales for different application scenarios to achieve a balance between lightweight and high performance. Focusing on the efficient detection requirements in industrial scenarios, YOLOv6 [29] introduces the Efficient Rep Backbone and an improved Rep PAN structure, significantly improving the inference speed and detection accuracy of the model, especially suitable for practical application scenarios such as edge devices and autonomous driving. In addition, YOLOv6 has optimized the end-to-end training process of the model, further reducing deployment costs. In the same year, Wang et al. proposed YOLOv7 [30], which adopted a novel Extended Efficient Layer Aggregation Network (E-ELAN) structure to enhance the model's feature representation ability. Meanwhile, the introduction of the Dynamic Label Assignment mechanism effectively improves the detection accuracy and robustness in multi-target detection scenarios. YOLOv8 marks a new stage for the YOLO series algorithms, adopting an anchor-free detection mechanism that breaks free from the limitations of traditional anchor-based methods, simplifies model design, and improves detection flexibility. In addition, YOLOv8 supports multitasking, such as instance segmentation and pose estimation, further expanding the application scope of

the algorithm. Thanks to a more efficient network structure, YOLOv8 maintains high accuracy while significantly improving reasoning speed, which is suitable for mobile terminals and edge computing devices. The subsequent updated detection model [31–33] serves as an iterative version of the YOLO series, aiming to further improve detection performance, while RT-DETR [34] utilizes Transformer architecture to achieve end-to-end real-time detection.

In recent years, advanced perception paradigms such as point cloud 3D detectors based on LiDAR (e.g., PointPillars [35], CenterPoint [36]) and bird's-eye view architectures based on cameras (e.g., BEVFormer [37]) have made significant progress in the field of autonomous driving. Although these frameworks can construct robust 3D spatial representations through multimodal or multi-view temporal Transformers, they usually encounter the challenges of large parameter quantities and high computational costs.

In the field of lightweight vehicle target detection, recent academic progress has been remarkable. For instance, Zhang et al. [38] introduced the C3k2_PS module into the YOLOv11 model, enhancing the feature interaction and global context modeling capabilities, effectively balancing the detection accuracy and inference efficiency. Regarding unmanned aerial vehicle remote sensing images, Zhang et al. [39] integrated the Mix-Mamba module into YOLOv12, optimizing the prediction performance of tiny vehicles in difficult samples. Additionally, Huang et al. [40] pointed out that although real-time end-to-end detectors (such as RT-DETR) perform well, they often lack sufficient detail expression capabilities when deployed in a lightweight manner, and still have limitations in the accuracy of small target detection. However, existing methods often struggle to balance complex background interference resistance and precise bounding box regression without significantly increasing the computational burden in extremely lightweight network architectures. Therefore, this paper addresses the pain points of vehicle target detection in complex traffic scenarios, where small targets are prone to missed detection and false detection due to large scale variations and mutual occlusion. We propose a lightweight vehicle detection model (MLRP-YOLOv8n) based on YOLOv8n to improve the robustness and accuracy of vehicle detection while maintaining extremely low computational costs.

2. Related Work

YOLOv8, as another iterative update of the YOLO series object detection framework, has improved detection accuracy while maintaining real-time performance through systematic architecture innovation. YOLOv8 continues the design concept of YOLO scalability, forming a complete model lineage covering YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), and YOLOv8x (extra-large) based on differences in depth and width parameters, with its model structure shown in Figure 1, can adapt to multi-scenario requirements from edge devices to cloud servers.

Its technological innovation is mainly reflected in three dimensions:

Enhanced feature engineering architecture: Optimizing gradient flow propagation through the C2f feature extraction module and cross stage local connections, which improves the training efficiency and feature expression ability of the model; combining a lightweight bidirectional Feature Pyramid Network to achieve multi-scale feature fusion, enhancing small object detection capability while maintaining computational efficiency.

Decoupling detection head optimization: Decoupling classification and regression tasks into independent prediction branches, using a task aligned label assignment strategy (Task Aligned Assign) and a soft weighting mechanism to coordinate the matching relationship between spatial priors and semantic confidence, significantly improving the efficiency of positive sample screening and prediction accuracy.

Training strategy optimization: Introducing an anchor-free detection paradigm to eliminate the inductive bias of preset anchor boxes. Adopting Dynamic NMS and loss

function improvement, gradually optimizing the weight allocation of difficult and easy samples through curriculum learning strategy.

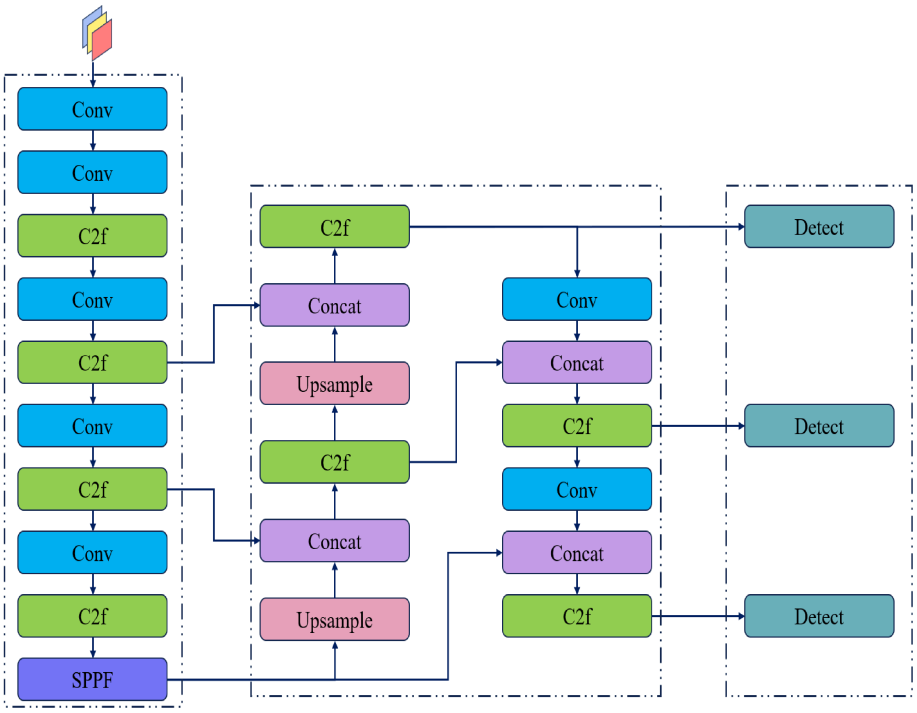


Figure 1. YOLOV8 network structure.

YOLOv8n is a lightweight single-stage detector that includes four core modules:

Input end: Adopting adaptive Mosaic data augmentation to enhance the model's generalization ability.

Backbone network: Extracting features through C2f modules and utilizing a dual path structure to achieve cross stage feature reuse.

Neck network: Using a bidirectional feature pyramid (FPN + PAN), integrating shallow spatial features and deep semantic features.

Detection head: Adopting a decoupling design to separate classification and regression tasks, reducing inter-task interference.

3. Methodology

3.1. Improved Feature Extraction Module Based on MLCA

An attention mechanism is a key technology in computer vision to enhance model expression ability. Its core idea is to strengthen important features and suppress redundant information through dynamic weight allocation. However, traditional attention mechanisms have two major limitations: channel attention (e.g., SE, ECA) only extracts channel statistics through global average pooling, ignoring local feature distributions in spatial dimensions, resulting in the loss of fine-grained information. Although hybrid attention mechanisms (e.g., CBAM and CA) attempt to integrate channel and spatial information, their computational complexity is relatively high.

Mixed Local Channel Attention (MLCA) [41] is a lightweight attention mechanism designed to enhance the expressive power and accuracy of object detection networks by integrating local spatial information and global channel information. Compared with traditional attention mechanisms, MLCA attempts to overcome the limitations of channel attention relying only on global channel statistics and ignoring spatial features, and solves the problem of high computational complexity and difficulty in adapting to lightweight

networks in hybrid attention mechanisms. Through this innovative design, MLCA can effectively balance performance and computational complexity, enabling the network to enhance its perception and processing capabilities of targets without significantly increasing computational burden, thereby improving the overall performance of the object detection system. Its structure is shown in Figure 2.

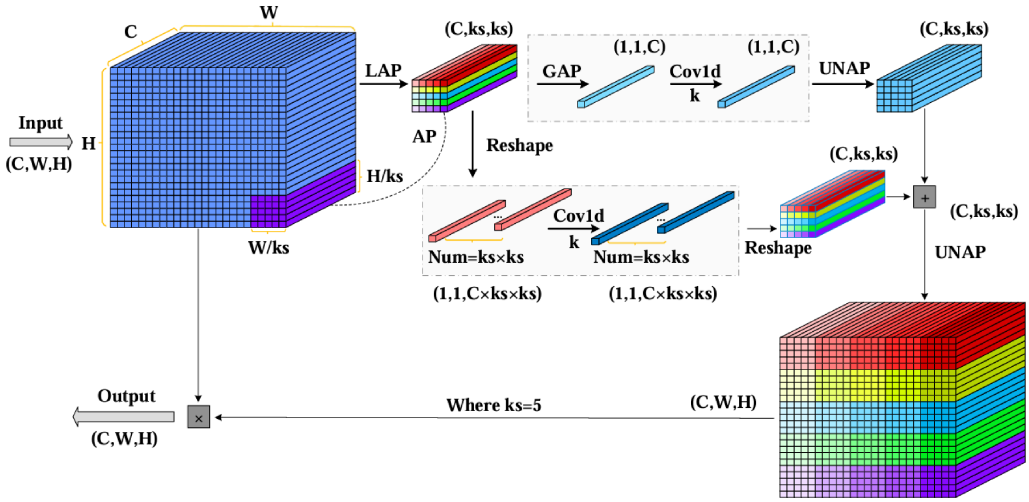


Figure 2. Mixed Local Channel Attention structure.

MLCA achieves a balance between performance and complexity through the following innovative designs:

(1) Local spatial information extraction

The input feature map is first subjected to Local Average Pooling (LAP), which divides the feature map into multiple local regions (e.g., 5×5 blocks), and each region is independently average pooled to generate a feature vector containing local spatial information. For example, when the input feature map size is $C \times H \times W$, LAP converts it to a dimension of $C \times ks \times ks$.

Among them, the value of k is determined by Formula (1):

$$k = \Phi(c) = \left| \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right|_{odd} \quad (1)$$

C represents the number of channels, k is the size of the convolution kernel, b and r are hyperparameters, odd represents the value of k, and if the calculated value of k is even, add 1 to make it odd.

(2) Dual branch information fusion

Global branch: perform global average pooling (GAP) on the input feature map to generate global channel statistics (dimension $1 \times 1 \times C$).

Local branching: Further processing of local features after LAP, capturing cross-channel local dependencies through Conv1D to avoid information loss caused by channel dimensionality reduction.

(3) Anti-Pooling and Weighted Fusion

The global and local branches are restored to their original resolution through UNAP, and then directly added and fused. This process preserves the weight distribution of local regions while incorporating the importance of global channels, ultimately generating attention weights through the Sigmoid function for recalibrating the input feature map.

The specific implementation of GAP, LAP, and UNAP operations is shown in Figure 3.

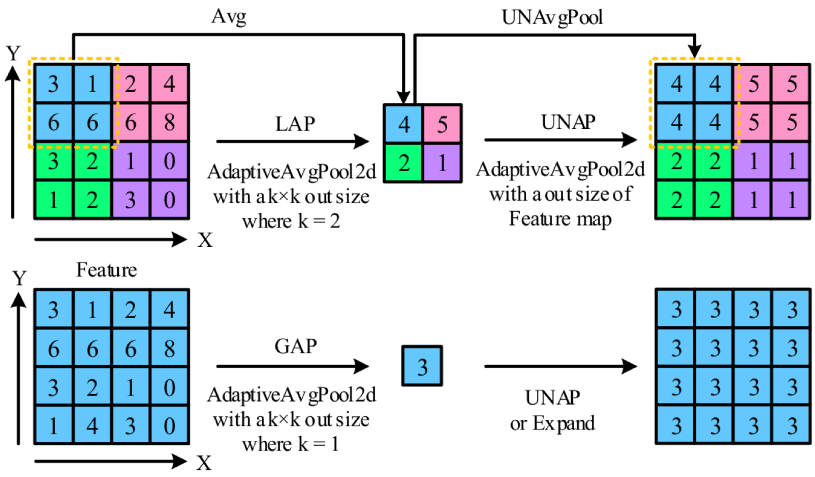


Figure 3. GAP LAP UNAP schematic diagram.

Overall, the MLCA mechanism achieves a balance between performance and complexity through innovative design. Firstly, Local Average Pooling (LAP) extracts fine-grained local spatial information to ensure that the model focuses on key regions. Secondly, dual branch information fusion combines global and local features, avoiding information loss. At the same time, anti-pooling and weighted fusion reconstructed the original feature map and generated accurate attention weights, thereby enhancing the model's attention to key information.

In the backbone network of YOLOv8n, the C2f module serves as the core part of feature extraction, enhancing the model's representation ability through the fusion of deep and shallow features and the use of cross-level feature connectivity. However, relying solely on convolution and concatenation operations cannot fully focus on "key channels" or "important local regions". Therefore, to further enhance the attention of C2f module to features, MLCA module is added to it. By dynamically adjusting channel weights during the Bottleneck stage, while considering both local spatial information and global channel information, the network's ability to focus on regions or features directly related to object detection during feature extraction is enhanced. The C2f structure after adding MLCA is shown in Figure 4.

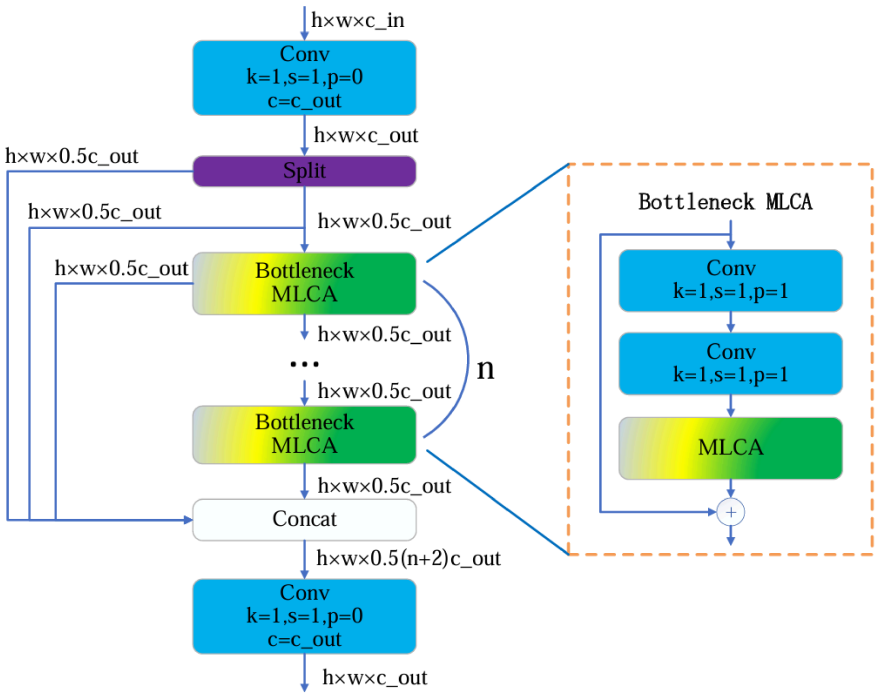


Figure 4. C2f structure after adding MLCA.

3.2. SPPF Structure Improved Based on LSKA

In convolutional neural networks, using a large kernel can directly expand the receptive field, allowing the network to capture a larger range of spatial information, which helps to learn the global features of the image. This ability is particularly important in tasks such as object detection and semantic segmentation, as it can better model long-range dependencies and enhance the network's understanding of complex patterns.

However, despite the expansion of receptive fields and the ability to extract global features brought by large convolutional kernels, it also presents computational and optimization challenges. Firstly, the computational cost and memory usage have significantly increased. Because large convolution kernels require processing larger input regions, the computational complexity involved in each convolution operation increases, resulting in longer training and inference times for the model. Secondly, during the optimization process, due to the expansion of the receptive field, the information transmission between convolutional layers becomes more complex, which may increase the risk of gradient vanishing or exploding, affecting the convergence speed and stability of the model.

Large Separable Kernel Attention (LSKA) [42] effectively solves the problem of large convolution kernels and fully utilizes the advantages of large kernel convolution. LSKA is an improved version of the large kernel attention (LKA) module, aimed at addressing the computational efficiency issues that LKA faces when dealing with large-sized convolution kernels.

LSKA adopts a core optimization strategy: Decomposing the 2D deep convolution kernel into cascaded 1D horizontal convolution ($1 \times k$) and 1D vertical convolution ($k \times 1$). This decomposition strategy enables feature extraction to be performed in both horizontal and vertical directions, reducing computational complexity from O (k^2) to O (k). Meanwhile, LSKA retains the channel adaptability and spatial adaptability of LKA, and is compatible with dilated convolution design, further enhancing the ability to model long-range dependencies.

In the YOLOv8 network, the Spatial Pyramid Pooling Fast (SPPF) module expands the receptive field through multi-scale max pooling. However, pooling operations can lead to the loss of feature information and have limited modeling capabilities for long-range dependencies. To address these issues, the LSKA mentioned above is added to the SPPF module to enhance the network's ability to extract target features. The structure of the improved SPPF module is shown in Figure 5. After multi-scale pooling, LSKA is used to simultaneously perceive the original fine-grained features and the coarse-grained features after three rounds of pooling. By utilizing the features of different receptive fields, adaptively enhance the feature response of important regions and suppress redundant information. Better integrate multi-scale features to enhance the robustness of the model to target scale changes and complex backgrounds.

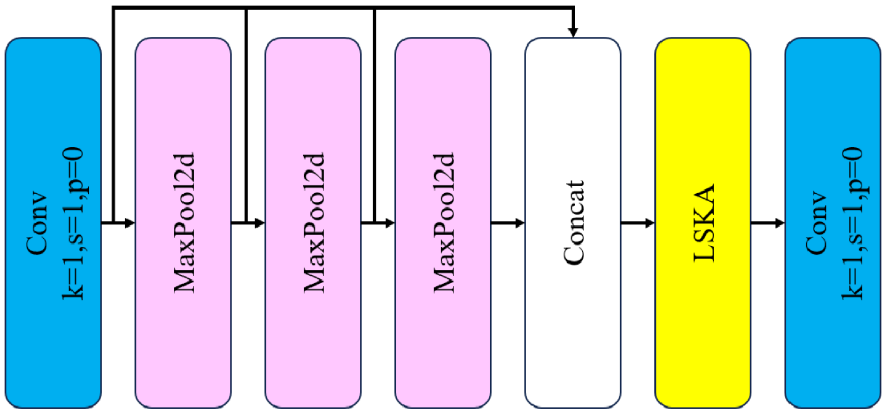


Figure 5. Schematic diagram of SPPF-LSKA.

3.3. Introduction of RFCBAM Convolution

RFCBAMConv is an improvement of the Convolutional Block Attention Module (CBAM) based on the RFA concept, and its structure is shown in Figure 6.

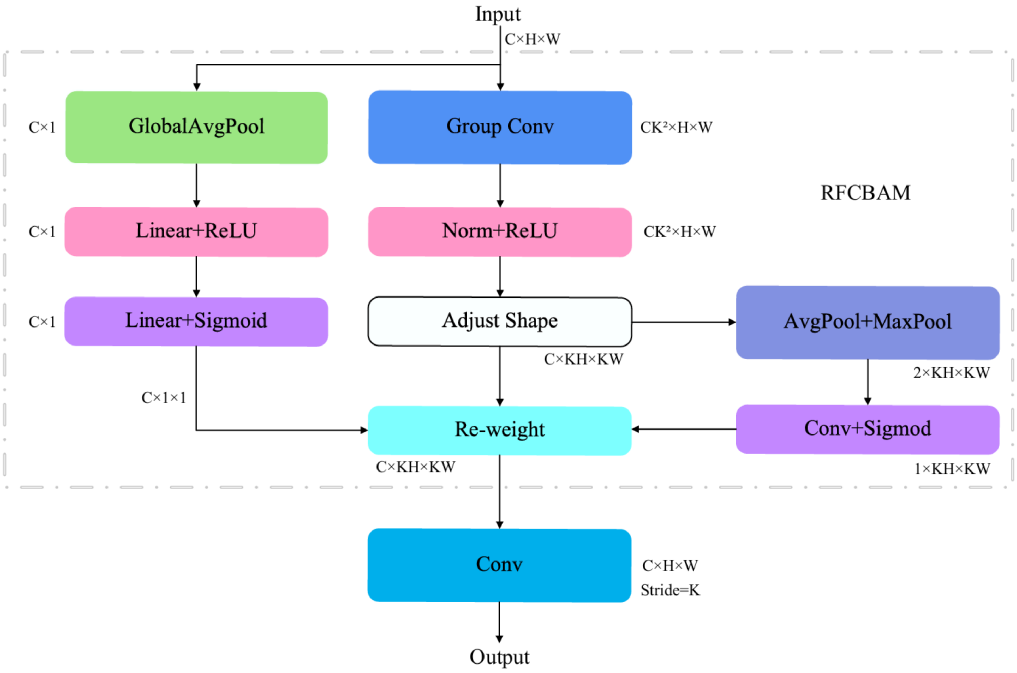


Figure 6. Schematic diagram of RFCBAMConv structure.

The original CBAM consists of a combination of channel attention and spatial attention, with its spatial attention module generating a two-dimensional attention map through global pooling and weighting the entire feature map. However, this approach does not differentiate the importance of local features within different receptive fields. RFCBAMConv limits the scope of spatial attention to a single receptive field, and dynamically generates attention weights for each receptive field to address the limitations of parameter sharing in large convolutional kernels.

In the original CBAM, channel attention and spatial attention were independently calculated, which may lead to insufficient information exchange. RFCBAMConv combines the two and synchronously weights the channel and spatial dimensions on the receptive field spatial features, enhancing the representational ability of local features. In addition, RFCBAMConv replaces the Channel Attention Module (CAM) in the original CBAM with the Squeeze and Excitation (SE) module, further reducing computational complexity.

By combining the attention weights of RFA with the convolution kernel parameters, RFCBAMConv generates dynamic non-shared convolution parameters, making the convolution operations within each receptive field unique and more accurate in adapting to local feature changes.

Considering the gain and controllable computational cost increase brought by RFCBAMConv for network feature extraction, this paper replaces the standard convolution in the neck part of YOLOv8 with RFCBAMConv to enhance the network's ability to extract target features.

3.4. Introduction of PIoUv2 Loss Function

The loss function, as feedback used in YOLO network models to measure the difference between the predicted output of the model and the true target, has a certain and substantial impact on the performance of the model. Therefore, the reasonable design

of the loss function not only determines whether the model can converge quickly and stably, but also directly affects the detection accuracy and generalization performance of the model in practical complex scenarios, which is an important guarantee for improving the performance of YOLO networks.

The loss function of YOLOv8 continues the classic multitasking approach of the YOLO series, consisting of three parts: confidence loss, classification loss, and bounding box regression loss.

Confidence loss is mainly responsible for determining whether there is a target object of a predetermined category within the prediction box. It uses binary cross entropy loss to dynamically adjust the weights of positive and negative samples, especially to enhance the attention to high-quality prediction boxes, thus solving the problem of serious imbalance in the number of positive and negative samples in object detection.

The role of classification loss is to help the model determine which categories the detected target belongs to. YOLOv8 uses multi-label binary cross entropy, allowing targets to belong to multiple categories simultaneously. At the same time, this loss will automatically adjust the weight of classification loss based on the degree of consistency between the predicted box and the real box, promoting the coordination and unity of classification and localization tasks.

The bounding box regression loss directly affects the accuracy of the model in predicting the target position. YOLOv8 combines CIoU loss and Distribution Focus Loss (DFL) in this stage, where CIoU comprehensively considers basic factors such as the overlapping area of the frame, the distance between the center points, and the aspect ratio, and can achieve preliminary optimization of the target position.

However, in complex scenarios, the CIoU loss still has limitations in its ability to perceive changes in box shape and scale: Firstly, the aspect ratio penalty term depends on angle calculation, making it difficult to constrain the size differences in linear proportional changes. Secondly, the normalization method does not differentiate between the independent optimization requirements of width and height. These deficiencies indicate that relying solely on CIoU is difficult to meet the detection needs of multi-scale, high deformation targets. Therefore, in response to the limitations of CIoU, it is necessary to combine its design principles with optimization ideas of other improved loss functions, and provide theoretical support for constructing more robust bounding box regression losses through comparative analysis of overlapping region penalties, shape constraint mechanisms, and multi-scale adaptability.

PIoU [43] is a good solution to the existing problem. The PIoU loss function links the penalty term to the size of the real box, and no longer uses the minimum closed bounding rectangle constructed based on the predicted box and the real box. It also designed more reasonable function forms and non-monotonic functions, successfully eliminating the drawbacks of blind expansion of prediction boxes and better balancing the gradient allocation of low-, medium-, and high-quality samples, thereby significantly accelerating convergence and improving accuracy. The calculation diagram is shown in Figure 7.

The PIoU loss function can be divided into two versions, where the calculation formula for the PIoUv1 loss function is shown in Formulas (2) and (3):

$$L_{PIoU_{v1}} = L_{IoU} + 1 - e^{-P^2},\ 0 \leq L_{PIoU} \leq 2 \tag{2}$$

$$P = \left(\frac{dw_1}{W_{gt}} + \frac{dw_1}{W_{gt}} + \frac{dh_1}{H_{gt}} + \frac{dh_2}{H_{gt}}\right)/4 \tag{3}$$

In the above equation, dw_1 represents the absolute value of the horizontal distance between the right edge of the real box and the predicted box, dw_2 represents the absolute

value of the horizontal distance between the left edge of the real box and the predicted box, dh_1 represents the absolute value of the horizontal distance between the upper edge of the real box and the predicted box, dh_1 represents the absolute value of the horizontal distance between the lower edge of the real box and the predicted box, and P is the penalty factor.

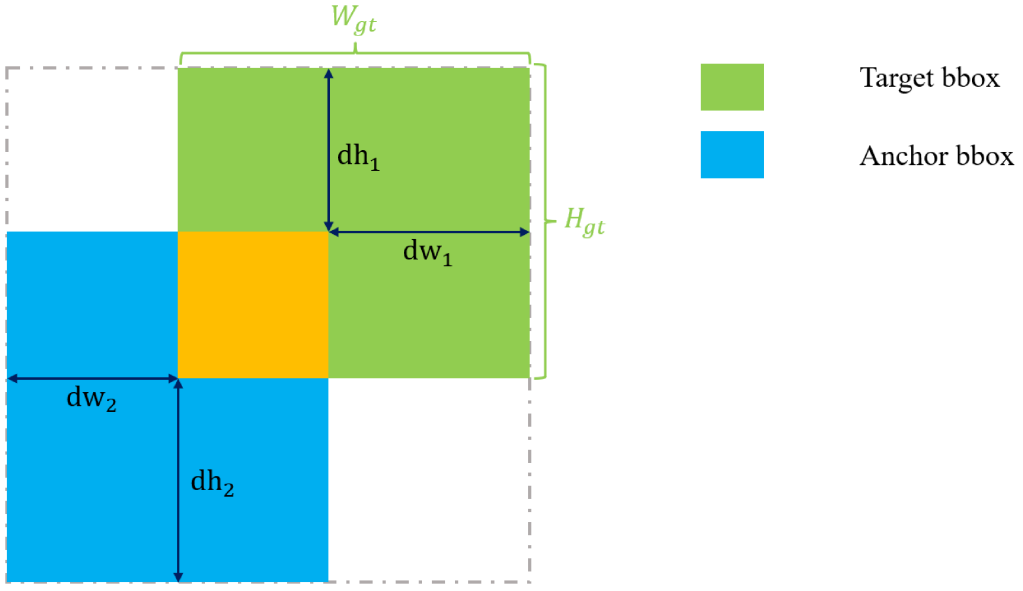


Figure 7. Schematic diagram of L_{PIoU} calculation.

PIoUv2 further adds a non-monotonic function based on PIoUv1, which can obtain different degrees of attention to the predicted boxes in different quality intervals, thereby further improving the accuracy of the detector. Its implementation can be represented by Formulas (4)–(6).

$$q = e^{-P}, q \in (0, 1] \tag{4}$$

$$\mu(x) = 3x \cdot e^{-x^2} \tag{5}$$

$$L_{PIoU_{v2}} = \mu(\lambda q) L_{PIoU} \tag{6}$$

As the denominator of the penalty factor P uses the width and height W_{gt} and H_{gt} of the real box, when the target size increases, a relatively small normalization penalty will be given to the same prediction error. When the target size decreases, it becomes more sensitive to the same prediction error. Only one hyperparameter λ was used, making parameter tuning more convenient. Therefore, based on the above comprehensive analysis, PIoUv2 is introduced as a new loss function to replace the original CIoU loss function in YOLOv8.

3.5. MLRP-YOLOv8n

The improved YOLOv8n network structure is shown in Figure 8. Firstly, the changes to the backbone network were completed by adding MLCA to the C2f module and LSKA to the SPPF module. Then RFCBAMConv convolution was used to replace the convolution in the neck network, and finally the PIoUv2 loss function was used as the bounding box regression loss function of the network. By using the above improvement strategies, improvements have been made to YOLOv8. This article refers to the improved YOLOv8 model as MLRP-YOLOv8n.

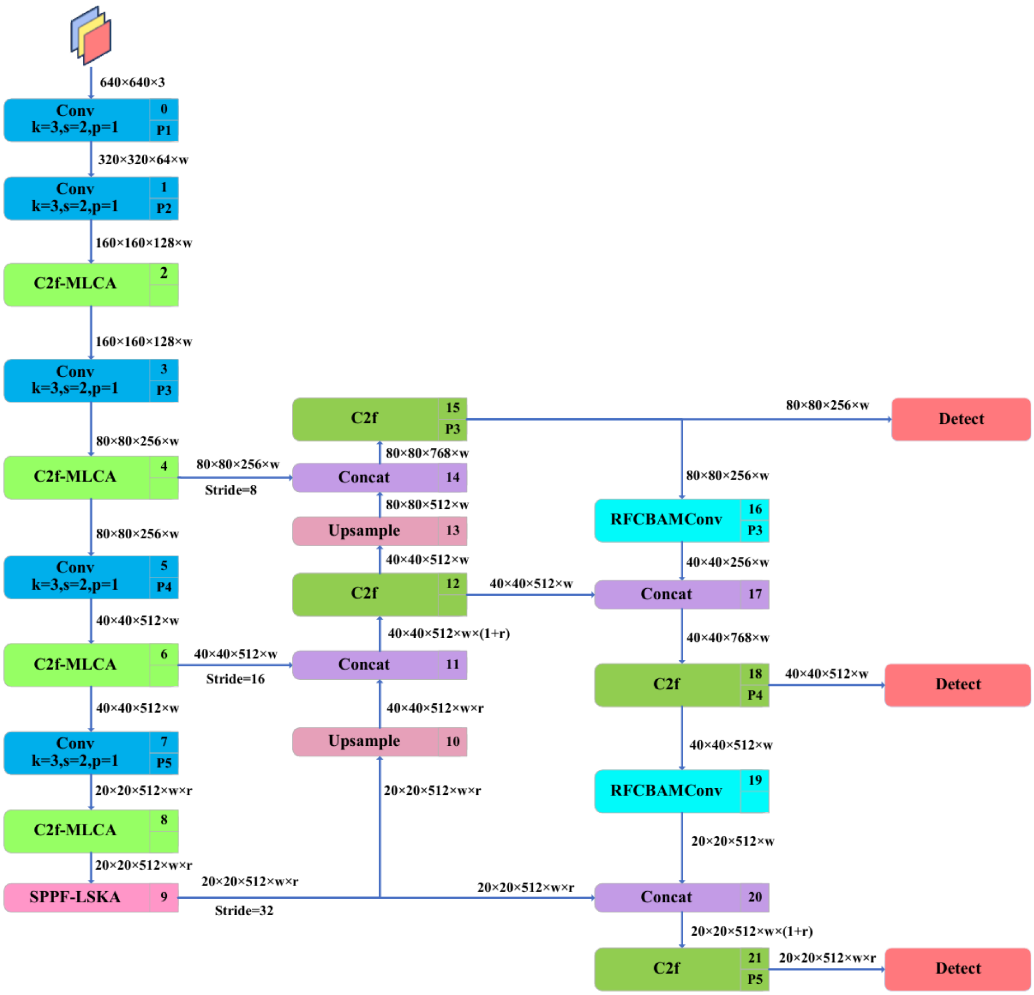


Figure 8. MLRP-YOLOv8n network structure.

4. Experiments

4.1. Dataset

The article utilized the authoritative public dataset KITTI [44] in the field of autonomous driving. As a benchmark dataset in the field of computer vision research, KITTI was jointly developed by the Karlsruhe Institute of Technology and the Toyota Technical Institute at Chicago, and is committed to providing a multi-dimensional evaluation benchmark for the environmental perception algorithm of the auto drive system.

This dataset collects real road scene data through a vehicle mounted multi-sensor platform, covering core tasks such as object detection, object tracking, stereo vision, and semantic segmentation. In the object detection task, the dataset contains a total of 14,936 images, including 7418 in the training set and 7518 in the testing set. Eight types of object detection, including Car, Pedestrian, Cyclist, Van, Truck, Tram, were constructed for complex traffic scenes.

This dataset addresses the common phenomena of target occlusion and truncation in autonomous driving scenarios. Based on the Visible Ratio and Truncation Level of the target's visible area, the samples are divided into three levels: easy (visible area > 80%), moderate (30% < visible area ≤ 80%), and hard (visible area ≤ 30%). This hierarchical design not only objectively reflects the progressive challenges of object detection tasks in actual road scenarios, but also provides fine-grained quantitative indicators for algorithm performance evaluation. Figure 9 shows a KITTI sample image.



Figure 9. KITTI object detection sample image.

Due to the lack of annotation information in the test set of KITTI dataset, 7481 training set images with annotation information were selected as experimental data in this chapter. Regarding the KITTI dataset, which did not use YOLO format for annotation and included non-vehicle target information in the target category, preprocessing of the data is required to adapt to subsequent model training and experimental analysis. First, we perform data cleaning on the dataset. Since our goal is vehicle recognition, we only retain the images in the dataset that contain information of the four categories: Car, Van, Truck, and Tram. We eliminate the images that do not contain these categories and remove the annotations of other categories. Then we convert the data annotation information and transform the coordinate system annotation into a YOLO recognizable annotation format. After data cleaning, it is randomly divided into training set, validation set, and testing set in a ratio of 8:1:1.

4.2. Experimental Configuration

To eliminate the interference of environmental variables on the experimental results, all experiments in this chapter were conducted on strictly controlled hardware platforms and software environments. The specific experimental environment configuration is detailed in Table 2.

Table 2. Experimental environment configuration.

Category	Name	Version	Quantity
Hardware	Operating System	Windows11	1
Hardware	GPU	RTX 3070	1
Hardware	CPU	i5-13490F	1
Hardware	Memory	16G	2
Software	Development environment	PyCharm Professional	-
Software	Programming language	Python 3.10.16	-
Software	Deep learning framework	PyTorch 2.2.2	-
Software	NVIDIA driver	560.94	-
Software	CUDA	12.6	-
Software	CUDANN	8.8.1	-

In order to ensure the objectivity and reproducibility of the experiment, and avoid unnecessary influence on the experimental results, all model hyperparameter settings in the experiment are kept consistent, ensuring that the conditions for each experiment are the same, so that the performance differences in different models or methods can be compared, and reliable and consistent results can be obtained. The specific hyperparameter settings are shown in Table 3:

Table 3. Detailed table of hyperparameter settings.

Parameters	Parameter Value	Note
Height	640	Input image height
Width	640	Input image width
Batch Size	16	Batch size
Epochs	100	Number of training rounds
Works	8	Number of threads
Learning Rate	0.01	-
Momentum	0.973	-
Weight Decay	0.0005	-
Optimizer	SGD	-
Time	≈6 h	Training time
GPU	>95%	GPU utilization rate

4.3. Evaluation Indicators

Average precision (AP), mean average precision (mAP), Giga Floating Point Operations Per Second (GFLOPs), and model parameter count are used as evaluation metrics for the improved algorithm.

Among them, mAP is one of the commonly used evaluation indicators in object detection, used to measure the average detection accuracy of algorithms in multiple categories. It is the arithmetic mean of the average precision of each category, which can comprehensively reflect the performance of the algorithm in different categories. AP is obtained by calculating the precision at different recall rates. Formula (7) is as follows:

$$AP = \int_0^1 Precision(r)\mathrm{d}r \tag{7}$$

Among them, $Precision\ (r)$ is the accuracy given a recall rate r. AP is obtained by calculating the area under the precision and recall curves. mAP is the average of all categories of AP, calculated using Formula (8) as follows:

$$mAP = \frac{1}{C}\sum_{i=1}^{C} AP_i \tag{8}$$

Among them, C is the total number of categories, and AP_i is the average accuracy of the i-th category. The higher the mAP value, the better the overall detection performance of the model, which can better identify targets of different categories.

Giga Floating Point Operations Per Second (GFLOPs) are important indicators for measuring the computational complexity and inference efficiency of a model, especially during the inference phase. GFLOPs represent the number of floating point operations performed by a model per second and are typically used to evaluate the computational complexity and real-time performance of the model. In object detection models such as

YOLO, GFLOPs can be obtained by calculating the floating-point computational complexity of the model. Formula (9) for calculating GFLOPs is:

$$\text{GFLOPs} = \frac{\text{Total Floating Point Operations}}{10^9} \tag{9}$$

Among them, Total Floating Point Operations is the sum of floating point operations in all layers during the forward propagation of the model. The smaller the GFLOPs value, the less computing resources the model requires.

The parameter size of a model represents the total number of trainable parameters, which is usually used to measure the complexity of the model and is closely related to its storage requirements and resource consumption during the training phase. The more parameters there are, the greater the complexity and storage requirements of the model. The fewer the number of model parameters, the lower the storage and computing requirements of the model, making it suitable for deployment in environments with limited computing resources.

By using these four evaluation indicators, the performance of object detection algorithms can be comprehensively evaluated. AP and mAP evaluate the detection accuracy of the algorithm, reflecting the recognition ability of the model on various types of targets; GFLOPs evaluate the inference efficiency of the model, reflecting the computational cost of the model in practical applications. The number of model parameters evaluates the complexity of the model and intuitively reflects the storage and computation requirements of the model. The comprehensive performance of the four can find a balance between accuracy and efficiency, thereby optimizing the design of the model.

4.4. IoU Comparison Experiment

In order to verify the differences between various loss functions, comparative experiments were conducted on training multiple IoU class loss functions on the dataset. As CIoU has been improved on the basis of GIoU and DIoU, CIoU was directly selected as the benchmark for this experiment, and compared with three loss functions: EIoU, SIoU, and PIoUv2.

In the experiment, YOLOv8n was used as the baseline model, keeping the network structure consistent with other hyperparameters, and only replacing the bounding box regression loss function to compare its impact on the convergence curve of the model. The focus here is on observing the box loss curve on the training set, which can intuitively reflect the convergence speed and stability of the model in the bounding box regression task.

Figure 10 shows the variation curves of box loss with iteration times during the training process of various loss functions on the KITTI Detection dataset subset.

The experimental results show that all four loss functions exhibit a rapid downward trend during the initial training period and tend to stabilize after approximately 30 to 40 epochs. According to the illustrated results, the convergence speed of PIoUv2 is significantly faster than other loss functions. In addition, the final bounding box loss on the dataset shows that CIoU, EIoU, and SIoU are all higher than PIoUv2, reflecting that PIoUv2 has better performance in bounding box regression tasks, thereby significantly improving the convergence efficiency of the model. In comparison, the effectiveness of EIoU is relatively inferior, while the performance difference between CIoU and SIoU is not significant.

Overall, these experimental results are consistent with the theoretical analysis of the gradient balance, convergence speed, and final regression accuracy of each loss function, further confirming the effectiveness of improving the performance of the detection model based on PIoUv2.

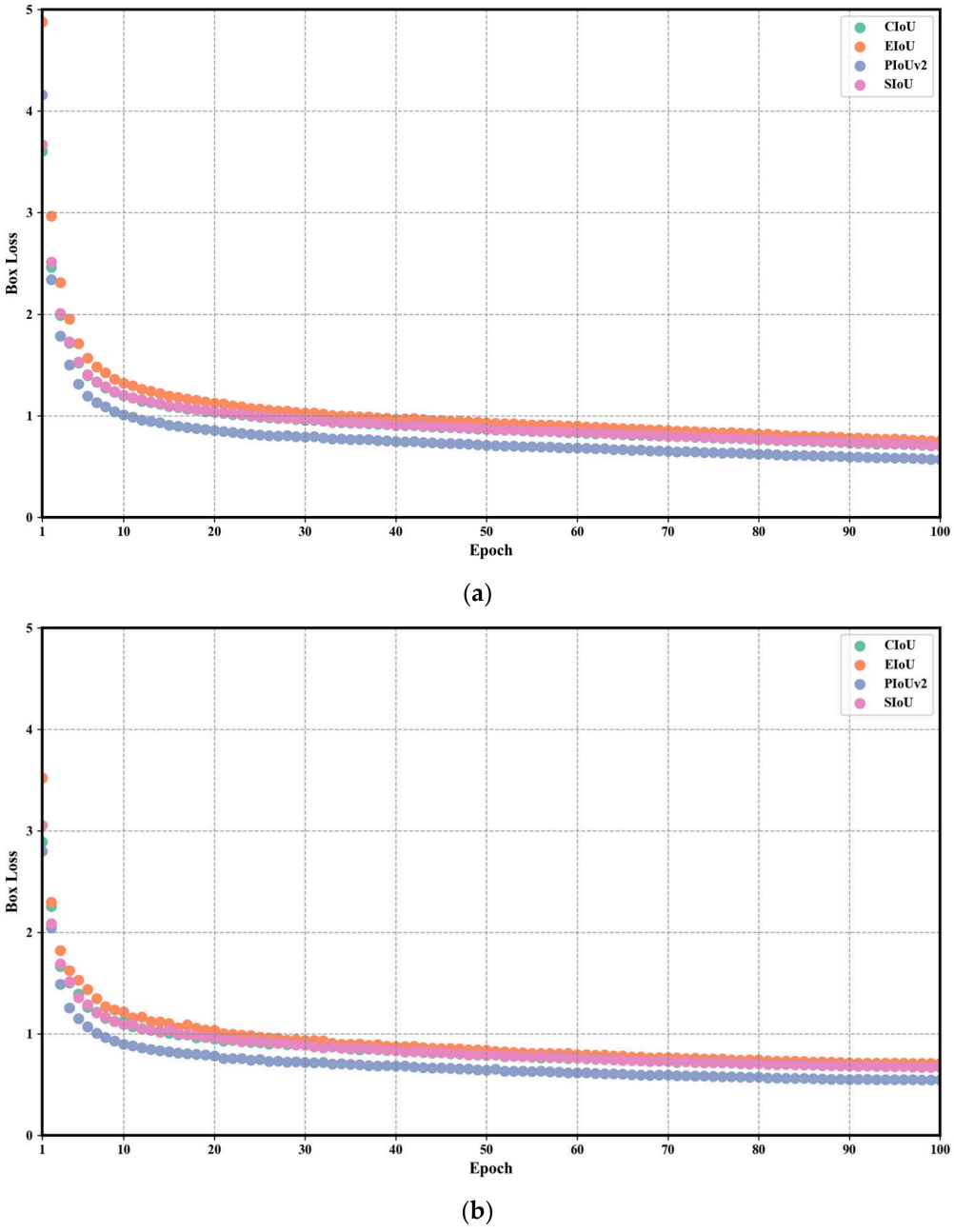


Figure 10. (**a**) Box loss comparison chart of the training set. (**b**) Box loss comparison chart of the validation set.

4.5. Ablation Experiment

The experiment used YOLOv8n as the baseline model and added four improvement strategies: MLCA, LSKA, RFCBAMConv, and PIoUv2 loss functions. Firstly, ablation experiments were conducted on the KITTI dataset subset to evaluate the specific impact of the newly introduced module on the model detection performance. The "$\sqrt{}$" represents the addition of this module, and the results of the ablation experiment are shown in Table 4. Among them, MA, LA, RC, and PI in the table represent MLCA, LSKA, RFCBAMConv, and PIoUv2, respectively. Car, Van, Truck, and Tram represent the percentage *AP* values of the model in this category.

Table 4. KITTI Detection dataset subset MLRP-YOLOv8 ablation experiment.

Model					Car	Van	Truck	Tram	mAP	GFLOPs	Params	FPS
v8n	MA	LA	RC	PI								
√	-	-	-	-	94.7	91.4	98.2	94.5	94.7	8.1	3.01 M	119
√	√	-	-	-	95.6	91.2	98.2	95.8	95.2	8.1	3.01 M	145
√	-	√	-	-	95.4	90.7	97.5	96.5	95.0	8.3	3.28 M	131
√	-	-	√	-	95.0	93.1	97.1	96.9	95.5	8.1	3.03 M	143
√	-	-	-	√	95.8	92.9	97.6	95.9	95.5	8.1	3.01 M	113
√	√	√	√	√	96.1	94.1	98.6	97.7	96.6	8.3	3.30 M	119

According to the ablation experiments in Table 4, it can be seen that after improving the model using MLCA, LSKA, RFCBAMConv, and PIoUv2, the detection performance of the model has been improved to varying degrees compared to the baseline model (YOLOv8n), proving that the addition of each module is effective in improving the model.

According to the results of the ablation experiment, the mAP of YOLOv8n was 94.7%, the GFLOPs were 8.1, and the parameter size was 3.01 M. After adding the MLCA module, compared with YOLOv8n, the mAP increased to 95.2%, an increase of 0.5 points. The detection accuracy of the Car and Tram categories both showed a certain degree of improvement, especially the Tram category with the largest increase of 1.3%. The GFLOPs and parameter count of the model did not significantly increase. This indicates that MLCA significantly improves detection accuracy by enhancing the model's ability to focus on key regions, without significantly introducing additional computational overhead.

After adding the LSKA module, mAP increased to 95% and achieved a 2% improvement in detection accuracy in the Tram category. But the GFLOPs of the model increased to 8.3 and the parameter count increased to 3.28 M. This is because LSKA optimized the detection ability of complex scenes through long-range dependency modeling, but also due to the increased computational complexity of attention, the computational load increased.

After adding RFCBAMConv, mAP increased to 95.5%, GFLOPs remained at 8.1, and the parameter count only slightly increased to 3.03 M. This achieved an improvement in detection accuracy for Car, Van, and Tram categories. This is mainly due to RFCBAMConv's integration of channel and spatial attention mechanisms, which better enhances the fusion ability of multi-scale features and comprehensively optimizes the detection performance of the model.

After replacing the original CIoU loss function with the PIoUv2 loss function, the accuracy of Car category detection reached 95.8%, an increase of 1.1%, Van category detection accuracy increased by 1.5%, Tram category detection accuracy increased by 1.4%, and mAP increased to 95.5%. The PIoUv2 loss function improves the bounding box regression loss function through a more reasonable design, which can guide the model to converge correctly and enhance its detection performance.

Although the introduction of the attention mechanism led to a slight decrease in inference speed compared to the baseline, the FPS of our MLRP-YOLOv8n on the NVIDIA RTX 3070 GPU remained above 100. This throughput significantly exceeded the industry standards for real-time performance in autonomous driving and vehicle cameras (30/60 FPS), fully demonstrating the practical feasibility of its deployment at the edge.

The ablation experiment results show that by adding different modules to YOLOv8n, the model performance has been improved to varying degrees, fully confirming the effectiveness of the model improvement strategies proposed in this paper.

It is worth noting that the final improved MLRP-YOLOv8n model exhibits a highly nonlinear and significant performance improvement compared to the baseline. This indicates that the four core components—C2f-MLCA, SPPF-LSKA, RFCBAMConv, and PIoUv2—form an organic and coordinated closed loop within the network, following a rigorous topological sequence: "feature decoupling and purification -> global scale expansion -> fusion of local enhancement -> pixel-level precise regression".

Firstly, in the backbone network (Backbone), the C2f-MLCA module enhances the expression fidelity of vehicle targets at the source of feature extraction by simultaneously modeling local spatial configuration and global channel correlation, locking in the basic boundaries and contour details. Subsequently, the SPPF-LSKA module located at the end of the backbone uses a large kernel attention mechanism to significantly expand the global receptive field, acting as a "scale amplifier" for features, mapping the high-quality local cues extracted by MLCA to multi-scale contexts. Finally, in the cross-scale feature fusion path of the neck network, the RFCBAMConv module introduces a crucial "local reinforcement" to the fused feature stream—it focuses on the key local regions within the receptive field, reactivating the vehicle local details (such as headlights, windshields, etc.) that might be diluted due to multiple layers of convolution and downsampling. This multi-level collaborative mechanism enables the model to output high-quality feature fields that balance the global context and microscopic details when facing multi-scale and highly occluded vehicles in the final stage.

Finally, from the perspectives of engineering reproducibility and computational efficiency, the proposed MLRP-YOLOv8n model was trained on a single NVIDIA RTX 3070 GPU, and it took approximately 6 h to complete 100 epochs. Throughout the training process, the hardware maintained a GPU utilization rate of over 95% that was stable and efficient. This indicates that although multiple refined improvement modules were introduced, the algorithm logic meticulously designed in this paper still has extremely high parallelism and pipeline optimization efficiency in hardware execution. There are no data loading bottlenecks or computational redundancies, fully aligning with the original design intention of the lightweight network.

4.6. Comparative Experiment

Firstly, in order to ensure the accuracy of the experiment, three independent runs were conducted using different random seeds to train the baseline model and the improved model. The results are shown in Table 5.

Table 5. Multiple experiments of YOLOv8n and MLRP-YOLOv8n.

	Type	First	Second	Third	Average
YOLOv8	Car	94.6	94.7	94.7	94.7
	Van	91.4	91.4	91.4	91.4
	Truck	98.2	98.2	98.3	98.2
	Tram	94.5	94.5	94.5	94.5
	mAP	94.6	94.7	94.7	94.7
MLRP-YOLOv8n	Car	96.1	96.1	96.1	96.1
	Van	94.1	94.0	94.1	94.1
	Truck	98.6	98.5	98.6	98.6
	Tram	97.6	97.7	97.7	97.7
	mAP	96.6	96.6	96.5	96.6

The obtained standard deviation is less than 0.05, indicating that the model training is relatively stable. For the comparative experiment conducted with the baseline model YOLOv8n, the result is the average value of multiple experiments. The specific evaluation indicators of the model detection are shown in Table 6 below.

Table 6. Comparison experiment between YOLOv8n and MLRP-YOLOv8n.

Model	Car	Van	Truck	Tram	mAP	GFLOPs	Params	FPS
YOLOv8n	94.7	91.4	98.2	94.5	94.7	8.1	3.01 M	147
MLRP-YOLOv8n	96.1	94.1	98.6	97.7	96.6	8.3	3.30 M	113

As shown in Table 6, the MLRP-YOLOv8n detection model proposed in this paper achieves a 1.6% performance improvement for AP in the Car category on the KITTI dataset compared to the YOLOv8n model. In the Van category, AP achieved a 2.7% performance improvement. In the Truck category, AP achieved a 3.3% performance improvement. In the Tram category, AP achieved a performance improvement of 3.2%. The mAP of the four target categories increased by 2.1 points, GFLOPs increased by 0.2, and the parameter count was 3.30 M.

Figure 11 shows the comparison of precision–recall (PR) curves between two models, from which it can be clearly seen that MLRP-YOLOv8n outperforms the original YOLOv8n model in all indicators.

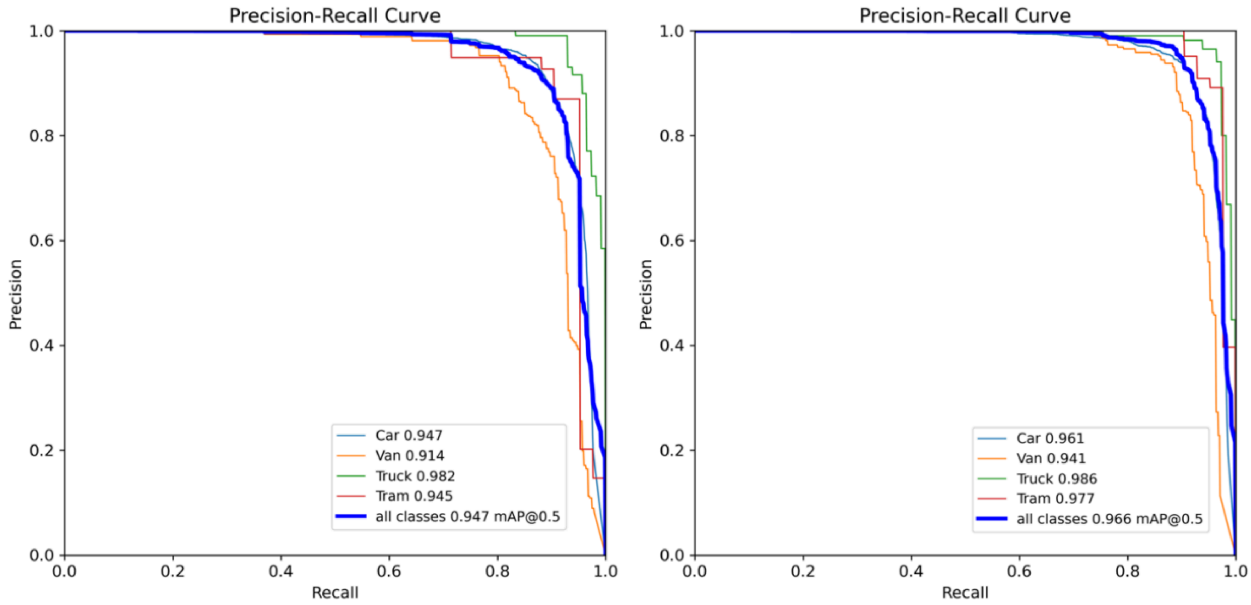


Figure 11. (**a**) The PR curve of YOLOv8n on the KITTI Detection dataset subset. (**b**) The PR curve of MLRP-YOLOv8n on the KITTI Detection dataset subset.

Area under the PR curve (AUC-PR), as an important indicator to measure the comprehensive performance of the model under sample imbalance conditions, achieved higher values on MLRP-YOLOv8n, indicating its stronger discriminative ability for positive and negative samples. Especially in the high recall range, MLRP-YOLOv8n can still maintain high accuracy, reflecting its stronger stability and robustness in detecting weak or edge targets. The balance improvement between precision and recall makes it more reliable and efficient in complex vehicle detection scenarios.

In order to demonstrate the detection performance before and after improvement more intuitively, the YOLOv8n model and the improved MLRP-YOLOv8n model were

used for detection in the same testing scenario. By comparing the detection results of the same image, we can see more clearly the differences between the two models in terms of false positives and false negatives. Figures 12 and 13 respectively show the detection results of two models for the same image, with YOLOv8n detection results on the left and MLRP-YOLOv8n detection results on the right.



(**a**) (**b**)

Figure 12. (**a**) The false detection performance of YOLOv8n on the KITTI Detection dataset subset. (**b**) The false detection performance of MLRP-YOLOv8n on the KITTI Detection dataset subset.



(**a**) (**b**)

Figure 13. (**a**) The missed detection effect of YOLOv8n on the KITTI Detection dataset subset. (**b**) The missed detection effect of MLRP-YOLOv8n on the KITTI Detection dataset subset.

By comparing the detection results of the same image, it can be found that YOLOv8n has a certain degree of false detection problem, which will determine nontarget objects as

vehicles, as shown in the red box area in Figure 12a. MLRP-YOLOv8n effectively solves these problems and can achieve accurate recognition of targets.

Meanwhile, by observing Figure 13, YOLOv8n also has the problem of missed detection, as it did not detect the target vehicle in the figure. However, MLRP-YOLOv8n has improved this missed detection problem by being able to identify vehicle targets and achieve accurate detection. Figure 13a uses a red box to mark the missed detection area.

To comprehensively and objectively evaluate the detection performance of the proposed model, a systematic comparative experiment was conducted between MLRP-YOLOv8n and current mainstream object detection models.

Table 7 summarizes the quantitative evaluation results of each model on various indicators, where the red font represents the optimal experimental results and the blue font represents the suboptimal experimental results. Figure 14 visualizes the detection performance of each model in four target categories, fully revealing the advantages and potential of the proposed model in practical applications. According to Table 7 and Figure 14, it can be seen that MLRP-YOLOv8n has achieved relatively leading detection accuracy in multi-class detection tasks. Among them, in the categories of Car, Truck, and Tram, AP reached 96.1%, 98.6%, and 97.7%, respectively, achieving the best detection performance. On the Van category, there is a small gap between the accuracy of 94.1% and RT-DETR (94.5%). Thanks to its performance in various subtasks, the overall mAP of MLRP-YOLOv8n reached 96.6%, which is approximately 1.2 percentage points higher than RT-DETR.

Table 7. Comparison experiment of KITTI Detection dataset subset (The red font indicates the optimal experimental result, while the blue font represents the suboptimal experimental result).

Model	Car	Van	Truck	Tram	mAP	GFLOPs	Params	FPS
RT-DETR	95.4	94.5	97.9	93.4	95.4	103.4	31.99 M	45
YOLOv3 Tiny	87.6	87.4	95.2	95.3	91.4	18.9	12.13 M	263
YOLOv5n	94.6	89.5	96.3	94.8	93.8	7.1	2.50 M	147
YOLOv6n	93.7	88.8	93.9	95.3	92.9	11.8	4.23 M	115
YOLOv7 Tiny	93.3	89.8	95.7	93.0	92.9	13.0	6.02 M	180
YOLOv9t	94.7	91.0	96.5	96.3	94.6	7.6	1.97 M	70
YOLOv10n	94.3	89.2	96.2	95.4	93.8	8.2	2.70 M	143
YOLO11n	93.9	89.9	95.7	92.9	93.1	6.3	2.58 M	123
YOLOv12n	94.3	90.6	97.1	94.3	94.1	6.3	2.56 M	130
MLRP-YOLOv8n	96.1	94.1	98.6	97.7	96.6	8.3	3.30 M	113

In terms of computational complexity and model size, MLRP-YOLOv8n only requires 8.3 GFLOPs and approximately 3.3 million parameters. Compared with RT-DETR (103.4 GFLOPs and 31.99 million parameters), the inference resource utilization is reduced by an order of magnitude. Compared to YOLOv3 Tiny (18.9 GFLOPs, 12.13 million parameters), it also has a higher efficiency and smaller model size. Apart from the RT-DETR model, the FPS values of all other models have exceeded the recognized real-time industry standards for autonomous driving and vehicle cameras. Some models such as YOLOv9t, YOLOv10n, YOLO11n, and YOLOv12n have achieved further lightweighting in architecture compared to MLRP-YOLOv8n. However, from the comparison of detection accuracy in the table, MLRP-YOLOv8n still has a significant advantage in accuracy performance, achieving a good performance balance between AP, mAP, computational overhead, and model parameter quantity, demonstrating excellent comprehensive performance.

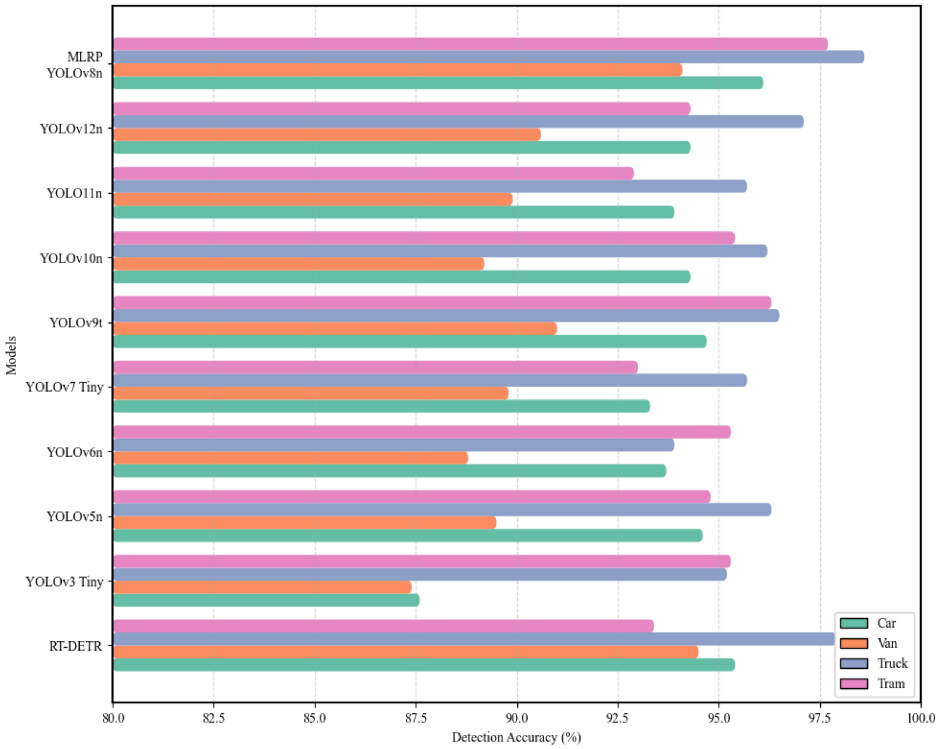


Figure 14. Visualization of indicator comparison analysis of KITTI Detection dataset subset.

4.7. Extended Experiment

To verify the generalization ability of the MLRP-YOLOv8n model, this paper uses the UA-DETRAC [45] dataset as a supplementary experiment. By preprocessing the dataset, a total of 140,000 images were obtained, including four annotation types: Car, Bus, Van, and Others. We extracted 10,000 images from them and divided the dataset in an 8:1:1 ratio.

In the previous KITTI Detection dataset experiments, MLRP-YOLOv8n demonstrated excellent detection capabilities. To further verify its robustness and stability in different scenarios, the PR curves of YOLOv8n and MLRP-YOLOv8n were also plotted on the UA-DETRAC dataset. The comparison results are shown in Figure 15.

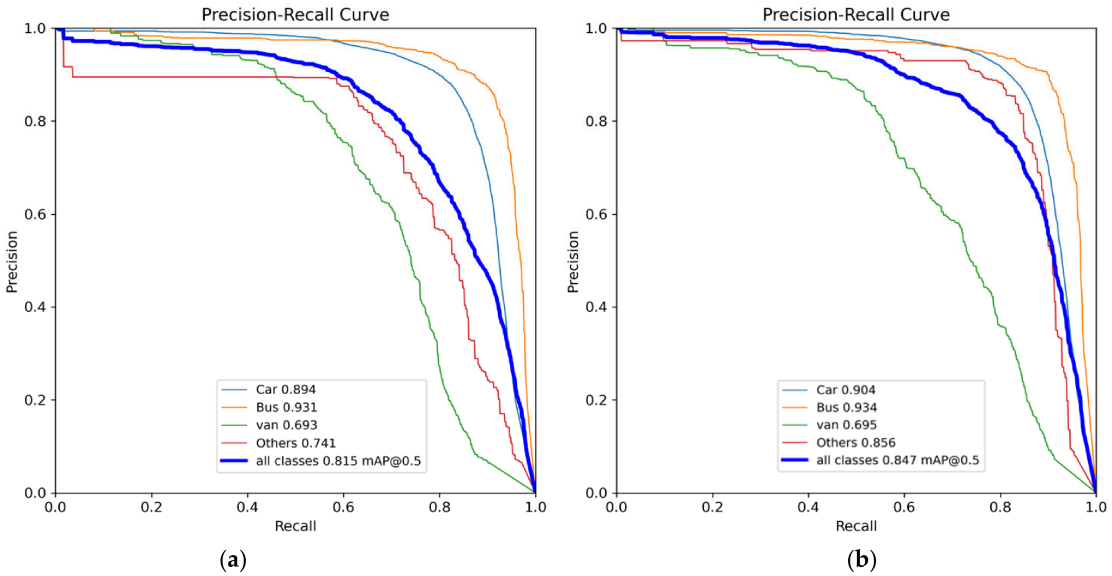


Figure 15. (**a**) The PR curve of YOLOv8n on the UA-DETRAC dataset. (**b**) The PR curve of MLRP-YOLOv8n on the UA-DETRAC dataset.

The specific detection results of MLRP-YOLOv8n on the UA-DETRAC dataset are shown in Figure 16. It covers various detection environments that may occur in actual vehicle target detection, such as sunny days, rainy days, cloudy days, and nighttime. The improved MLRP-YOLOv8n still demonstrates excellent target detection capabilities and can accurately identify targets in complex and variable environments.



Figure 16. Detection results of the UA-DETRAC dataset.

To compare the improvement effect of the model, some identical scenarios were selected from the test set to visually compare the detection results. The results are shown in Figure 17. By comparing these figures, in the case of a high density of vehicle targets, YOLOv8n has some issues of false detection and missed detection. For example, relatively large target categories such as "bus" cannot be accurately identified, or pedestrian targets are misjudged as vehicle targets.

On the UA-DETRAC dataset, the results of experiments comparing multiple models are shown in Table 8.

Table 8. UA-DETRAC dataset comparison experiment (The red font indicates the optimal experimental result, while the blue font represents the suboptimal experimental result).

Model	Car	Van	Truck	Tram	mAP	GFLOPs	Params	FPS
RT-DETR	88.6	90.5	68.7	70.0	79.4	103.4	31.99 M	55
YOLOv3 Tiny	87.6	92.3	68.9	83.2	83.0	18.9	12.13 M	267
YOLOv5n	89.7	92.5	71.6	79.2	83.3	7.1	2.50 M	145
YOLOv6n	87.8	92.5	64.5	65.0	77.5	11.8	4.23 M	160
YOLOv7 Tiny	88.1	91.6	67.6	65.2	78.1	13.0	6.02 M	203
YOLOv8n	89.4	93.1	69.3	74.1	81.5	8.1	3.01 M	155
YOLOv9t	89.6	93.1	70.1	78.5	82.8	7.6	1.97 M	70
YOLOv10n	88.8	92.2	70.1	74.8	81.5	8.2	2.70 M	145
YOLO11n	89.6	93.1	72.2	73.6	82.1	6.3	2.58 M	131
YOLOv12n	89.3	91.5	68.8	73.1	80.7	6.3	2.56 M	126
MLRP-YOLOv8n	90.4	93.4	69.5	85.6	84.7	8.3	3.30 M	111



(**a**) (**b**)

Figure 17. (**a**) The detection performance of YOLOv8n on the UA-DETRAC dataset. (**b**) The detection performance of MLRP-YOLOv8n on the UA-DETRAC dataset.

By comparing the experimental results in Table 8 with the visualization of the detection indicators in Figure 18, MLRP-YOLOv8n still maintains stable detection performance. However, RT-DETR, which performed well on the KITTI Detection dataset subset, did not achieve any detection advantage in this experiment despite having a larger computational cost and parameter count.

Among the four target categories, MLRP-YOLOv8n achieved the best detection accuracy in the Car, Bus, and Others categories. MAP is also superior to various mainstream detection models. Other lightweight models, such as YOLOv5n, have achieved good results on the UA-DETRAC dataset, but there is still a certain gap compared to MLRP-YOLOv8n. However, other lightweight models such as YOLOv9t, YOLOv10n, YOLO11n, and YOLOv12n still fail to achieve a balance between detection accuracy, model complexity, and computational overhead.

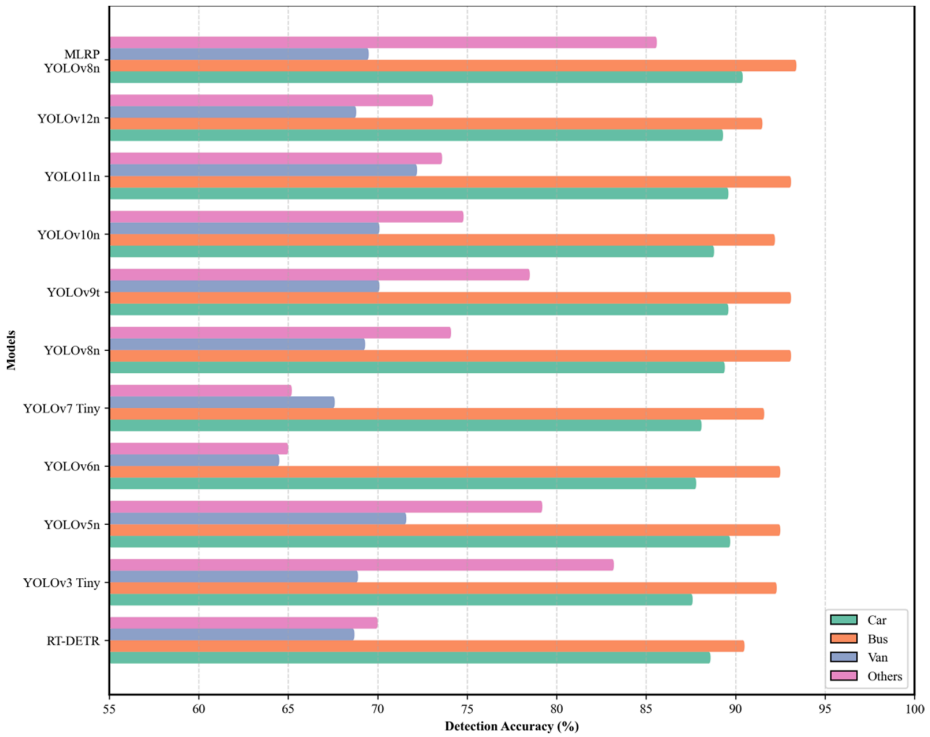


Figure 18. Visualization of indicator comparison analysis in UA-DETRAC dataset.

5. Discussions

Through ablation experiments on the KITTI Detection dataset subset and comparative experiments with other mainstream detection models, it is known that MLRP-YOLOv8n exhibits excellent detection performance in multi-target category detection. The effective balance between detection accuracy, computational cost, and model size is achieved, and the specific comparison visualization results of the three key indicators are shown in Figure 19.

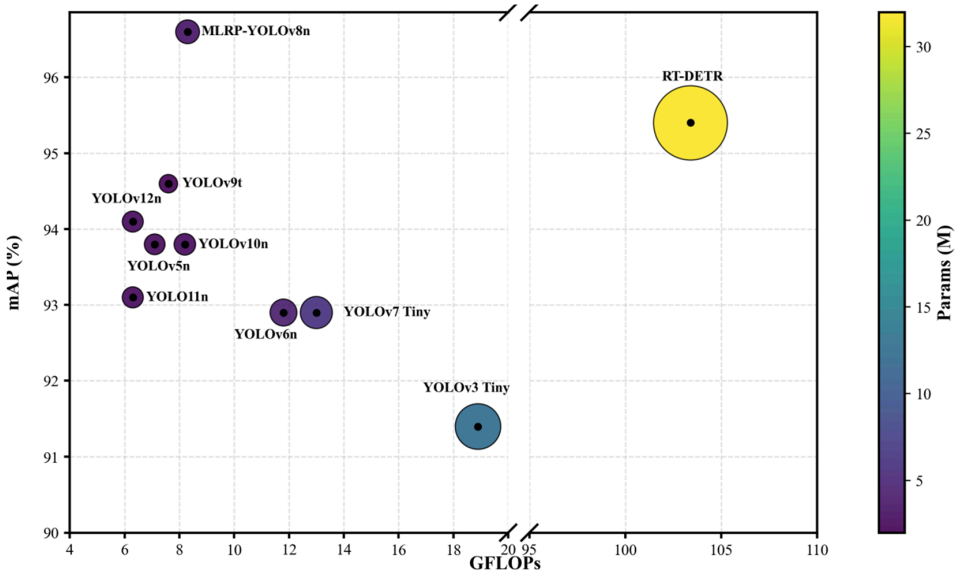


Figure 19. Comparative analysis of evaluation indicators for KITTI Detection dataset subset models.

From the comparison results shown in Figure 19, it can be further observed that MLRP-YOLOv8n exhibits comprehensive advantages of high detection accuracy, low computational cost, and small model scale in different detection models, significantly leading other YOLO detection models. Specifically, the model achieved 96.6% mAP with

only 8.3 GFLOPs and approximately 3.3 million parameters, which is far superior to other lightweight models such as YOLOv3 Tiny, YOLOv5n, YOLOv6n, and YOLOv7 Tiny. In terms of accuracy, it also surpasses the newly released YOLO series versions such as YOLOv9t, YOLOv10n, YOLO11n, and YOLOv12n.

These subsequent YOLO versions have undergone further lightweight design in their network architecture, resulting in a decrease in computational complexity. However, in the application of vehicle object detection, facing multiple targets and complex detection scenarios, the target discrimination ability has declined, and it is ultimately difficult to surpass MLRP-YOLOv8n in accuracy.

In contrast, MLRP-YOLOv8n significantly improves the information exchange efficiency between shallow and deep features by using an improved feature extraction module. While maintaining model compactness, it effectively enhances the modeling ability for small targets and complex backgrounds, thus achieving a good unity of detection accuracy and efficiency.

In addition, although RT-DETR performs similarly to our model in terms of detection accuracy, its model scale is as high as 103 GFLOPs and approximately 31.99 million parameters, significantly higher than MLRP-YOLOv8n. In practical applications, it has significant resource overhead and deployment burden. Furthermore, the experimental results indicate that RT-DETR is difficult to fully utilize its structural potential in scenarios with limited dataset size and few training rounds, resulting in a certain degree of decline in detection performance. Therefore, in comparison, MLRP-YOLOv8n has higher practicality and comprehensive competitiveness in practical application environments where resources are limited or rapid deployment is required.

The specific comparison visualization results of the three key indicators of detection accuracy, computational cost, and model size on the UA-DETRAC dataset are shown in Figure 20.

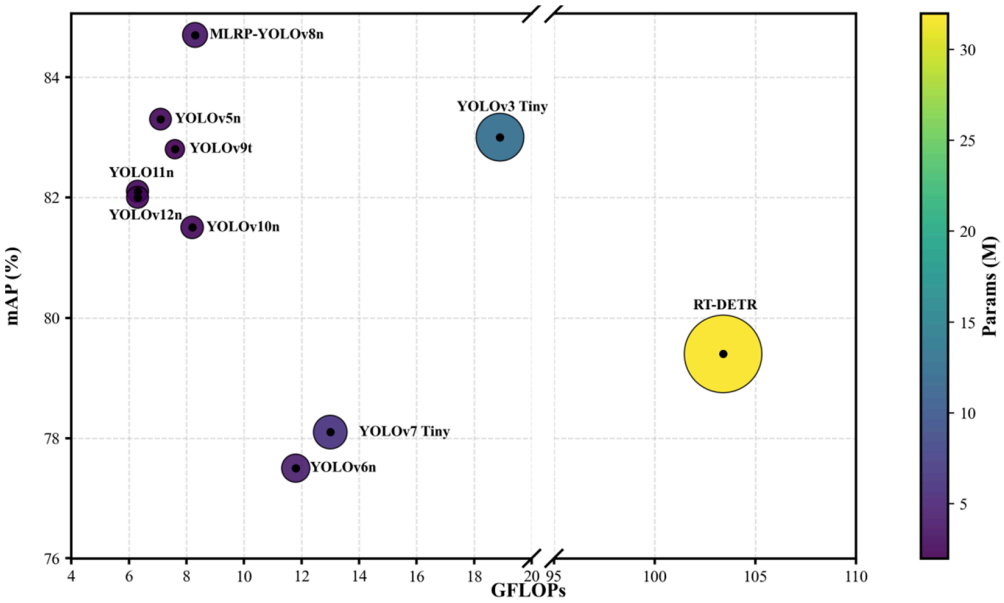


Figure 20. Comparative analysis of evaluation indicators for UA-DETRAC dataset models.

According to Figure 20, MLRP-YOLOv8n still exhibits a high detection accuracy advantage on the UA-DETRAC dataset. However, RT-DETR and other YOLO series models have experienced significant fluctuations in detection accuracy.

The comparative analysis between the KITTI Detection dataset subset and the UA-DETRAC dataset shows that the proposed MLRP-YOLOv8n model exhibits strong advan-

tages in detection accuracy and stability. This model not only achieves better detection performance on multiple key target categories, but also maintains high efficiency in terms of computational cost and model complexity, demonstrating good lightweight characteristics and practical deployment value. Overall, MLRP-YOLOv8n achieves an ideal balance between detection performance and resource consumption, with strong application potential and engineering adaptability.

6. Conclusions

Aiming at the high-precision detection requirements in the field of autonomous driving, the MLRP-YOLOV8n vehicle object detection algorithm is proposed. By embedding the MLCA mechanism into the C2f feature extraction module in the YOLOv8 backbone network, the network's ability to extract features from the target area is enhanced. Then, the LSKA mechanism is added to the SPPF module to integrate feature information of different granularities, adaptively enhance the feature response of important regions, better integrate multi-scale features, and improve the robustness of the model to target scale changes and complex backgrounds. Subsequently, RFCBAMConv convolution was used to replace the original convolution in the neck network, enhancing the cross-level correlation of multi-scale feature maps and improving small target missed detections. Finally, the PIoUv2 loss function is used instead of the original CIoU loss function. By utilizing its introduced non-monotonic focusing mechanism, while maintaining the ability to model the geometric relationship of the target box, it enhances the attention to medium-quality anchor boxes, suppresses the gradient interference of low-quality samples, accelerates the convergence speed of the model, improves positioning accuracy, and effectively alleviates common regression errors and border expansion problems in target detection. The effectiveness of the model improvement was demonstrated through experimental analysis on the KITTI Detection dataset subset and UA-DETRAC dataset.

Although the proposed MLRP-YOLOv8n has achieved significant optimization in trade-offs, there are still some operational boundaries and structural costs that need to be addressed in the actual deployment mode.

Firstly, since our architecture operates entirely based on a two-dimensional image plane, its detection reliability may be challenged in extreme safety-critical scenarios (such as severe object occlusion or poor night visibility). Additionally, from a structural perspective, the significant accuracy advantage we have achieved is at the cost of a slight increase in computational complexity. There will be a slight increase in parameters and GFLOPs, which subsequently will have a slight delay impact on inference throughput.

To eliminate these limitations while retaining the ultra-lightweight advantage, our future research will be divided into two complementary paths. Firstly, to ingeniously solve the problem of computational expansion, we plan to implement post-training structural pruning and network quantization strategies. By mathematically eliminating redundant attention paths and compressing model weights, we expect to completely eliminate the hardware footprint of the newly introduced modules. Secondly, to overcome the inherent limitations of two-dimensional vision, we plan to study the cross-modal knowledge distillation paradigm, fully leveraging their spatial geometric structure and depth perception, which can implicitly transfer to our 2D network.

Author Contributions: Conceptualization, W.Y. and F.N.; methodology, W.Y. and S.Y.; validation, F.N.; resources, W.Y.; data curation, Q.Z.; writing—original draft preparation, W.Y.; writing—review and editing, S.Y. and F.N.; visualization, W.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Provincial and Ministerial Platform Intelligent Manufacturing Engineering Laboratory (PT2025KJS002) of Shandong Huayu University of Technology.

Data Availability Statement: The datasets presented in this study are available through https://www.cvlibs.net/datasets/kitti/ (accessed on 25 March 2026) and http://detrac-db.rit.albany.edu/ (accessed on 25 March 2026).

Acknowledgments: We would like to express our gratitude to Shandong Huayu University of Technology for providing the research platform, as well as the joint efforts of the cooperating universities.

Conflicts of Interest: The authors declare that they have no conflicts of interest.

References

1. Crayton, T.J.; Meier, B.M. Autonomous Vehicles: Developing a Public Health Research Agenda to Frame the Future of Transportation Policy. *J. Transp. Health* **2017**, *6*, 245–252. [CrossRef]
2. Liang, L.; Ma, H.; Zhao, L.; Xie, X.; Hua, C.; Zhang, M.; Zhang, Y. Vehicle Detection Algorithms for Autonomous Driving: A Review. *Sensors* **2024**, *24*, 3088. [CrossRef] [PubMed]
3. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
4. Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; Darrell, T. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020.
5. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014.
6. Uijlings, J.R.; Van De Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective Search for Object Recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [CrossRef]
7. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005.
8. Lowe, D.G. Object Recognition from Local Scale-Invariant Features. In Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV), Kerkyra, Greece, 20–27 September 1999.
9. Schuldt, C.; Laptev, I.; Caputo, B. Recognizing Human Actions: A Local SVM Approach. In Proceedings of the 17th International Conference on Pattern Recognition (ICPR), Cambridge, UK, 26 August 2004.
10. Neubeck, A.; Van Gool, L. Efficient Non-Maximum Suppression. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 20–24 August 2006.
11. Erhan, D.; Szegedy, C.; Toshev, A.; Anguelov, D. Scalable Object Detection Using Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
12. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014.
13. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
14. Girshick, R. Fast R-CNN. In Proceedings of the International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015.
15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
16. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
17. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-Cnn. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
18. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated Recognition, Localization and Detection Using Convolutional Networks. *arXiv* **2013**, arXiv:1312.6229.
19. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.
20. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single Shot Multibox Detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–10 and 15–16 October 2016.
21. Fu, C.-Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. Dssd: Deconvolutional Single Shot Detector. *arXiv* **2017**, arXiv:1701.06659. [CrossRef]

22. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
23. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015.
24. Redmon, J.; Farhadi, A. Yolov3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767. [CrossRef]
25. Law, H.; Deng, J. Cornernet: Detecting Objects as Paired Keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
26. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint Triplets for Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
27. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. Scaled-Yolov4: Scaling Cross Stage Partial Network. In Proceedings of the IEEE/cvf Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021.
28. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
29. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976. [CrossRef]
30. 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. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023.
31. Wang, C.-Y.; Yeh, I.-H.; Mark Liao, H.-Y. Yolov9: Learning What You Want to Learn Using Programmable Gradient Information. In Proceedings of the European Conference on Computer Vision (ECCV), Milano, Italy, 29 September–4 October 2024.
32. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Vancouver, WA, USA, 10–15 December 2024.
33. Tian, Y.; Ye, Q.; Doermann, D. Yolov12: Attention-Centric Real-Time Object Detectors. *arXiv* **2025**, arXiv:2502.12524.
34. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. Detrs Beat Yolos on Real-Time Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024.
35. Sun, L.; Li, Y.; Qin, W. PEPillar: A Point-Enhanced Pillar Network for Efficient 3D Object Detection in Autonomous Driving. *Vis. Comput.* **2025**, *41*, 1777–1788. [CrossRef]
36. Dong, H.; Zhang, H.; Chen, S.; Xin, J.; Zheng, N. CenterNext: Learning Generalized Gaussian Representation for LiDAR 3D Object Detection. In Proceedings of the 2025 IEEE 28th International Conference on Intelligent Transportation Systems (ITSC), Gold Coast, Australia, 18–21 November 2025.
37. Bao, X.; Liu, F.; Chen, Y.; Li, Y.; Tian, R. DMformer: A Transformer with Denoising and Multi-Modal Data Fusion for Enhancing BEV Perception. *Complex. Intell. Syst.* **2025**, *11*, 368. [CrossRef]
38. Zhang, X.; Jiang, H.; Han, D.; Guo, H.; Zhang, X.; Sun, K. A Method for Lightweight Pedestrian and Vehicle Detection for Unmanned Ground Vehicles in Open Environments. *Machines* **2026**, *14*, 527. [CrossRef]
39. Zhang, K.; Zhu, N.; Zhao, F.; Zhang, Q. MCViM-YOLO: Remote Sensing Vehicle Detection for Sustainable Intelligent Transportation. *Sustainability* **2026**, *18*, 2836. [CrossRef]
40. Huang, J.; Li, T. Small Object Detection by DETR via Information Augmentation and Adaptive Feature Fusion. In Proceedings of the 2024 ACM ICMR Workshop on Multimodal Video Retrieval, Phuket, Thailand, 10–14 June 2024.
41. Wan, D.; Lu, R.; Shen, S.; Xu, T.; Lang, X.; Ren, Z. Mixed Local Channel Attention for Object Detection. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106442. [CrossRef]
42. Lau, K.W.; Po, L.-M.; Rehman, Y.A.U. Large Separable Kernel Attention: Rethinking the Large Kernel Attention Design in CNN. *Expert Syst. Appl.* **2024**, *236*, 121352. [CrossRef]
43. Liu, C.; Wang, K.; Li, Q.; Zhao, F.; Zhao, K.; Ma, H. Powerful-IoU: More Straightforward and Faster Bounding Box Regression Loss with a Nonmonotonic Focusing Mechanism. *Neural Netw.* **2024**, *170*, 276–284. [CrossRef] [PubMed]
44. Geiger, A.; Lenz, P.; Urtasun, R. Are We Ready for Autonomous Driving? The Kitti Vision Benchmark Suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012.
45. Wen, L.; Du, D.; Cai, Z.; Lei, Z.; Chang, M.-C.; Qi, H.; Lim, J.; Yang, M.-H.; Lyu, S. UA-DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking. *Comput. Vis. Image Underst.* **2020**, *193*, 102907. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.