Journal of Sensor Technology and Application 传感器技术与应用, 2025, 13(3), 229-241
Published Online May 2025 in Hans. https://www.hanspub.org/journal/jsta
https://doi.org/10.12677/jsta.2025.133023

Hans汉斯

基于OSTD-YOLO复杂路况下的障碍物检测研究

赵晓卓，张成涛*，许季何，骆远鹏，黎俊宏

广西科技大学机械与汽车工程学院，广西 柳州

收稿日期：2025年3月13日；录用日期：2025年4月22日；发布日期：2025年5月9日

摘　要

随着自动驾驶技术的不断推进，在面对含有小目标、遮挡目标以及目标尺寸变化较大的复杂路况时，障碍物检测中的漏检与误检问题成为亟待攻克的关键难题。为此，设计了名为OSTD-YOLO (Occlusion and Small Obstacle Detection)的算法。该算法以YOLO11n作为基础模型展开构建，主要有两方面的改进。一是构建了增强小目标特征金字塔模块EMTFP (Enhanced Micro-Target Feature Pyramid)，此模块依托PAFPN架构，引入SPDConv对P2特征层进行精细化处理，有效提升了小目标特征提取能力。同时，受CSP思想和OmniKernel启发，研发出CSP-OmniKernel模块，用于整合不同层级提取的各类特征，让特征信息更加精炼。二是在模型输出端引入DyHead注意力机制检测头，实现了对空间、尺度和任务维度的自适应特征增强，全面提升了模型在复杂路况下对障碍物的感知精度。在公开数据集BDD 100k上进行测试后发现，相较于原始的YOLO11n模型，OSTD-YOLO算法的mAP50提升了近5个百分点，参数量为3.58 M，在与其他同类方法的对比试验中展现出参数量与精度的最佳平衡性。一系列试验充分表明，OSTD-YOLO算法能够出色地应对复杂路况下的障碍物检测任务，为自动驾驶技术的落地应用提供了有力支撑。

关键词

复杂路况，障碍物检测，深度学习，YOLO

Research on Obstacle Detection in Complex Road Conditions Based on OSTD-YOLO

Xiaozhuo Zhao, Chengtao Zhang*, Jihe Xu, Yuanpeng Luo, Junhong Li

School of Mechanical and Automotive Engineering, Guangxi University of Science and Technology, Liuzhou Guangxi

Received: Mar. 13th, 2025; accepted: Apr. 22nd, 2025; published: May 9th, 2025

*通讯作者。

文章引用: 赵晓卓, 张成涛, 许季何, 骆远鹏, 黎俊宏. 基于 OSTD-YOLO 复杂路况下的障碍物检测研究[J]. 传感器技术与应用, 2025, 13(3): 229-241. DOI: 10.12677/jsta.2025.133023

Abstract

With the continuous advancement of autonomous driving technology, in the face of complex road conditions with small targets, occluded targets and large changes in target size, the problem of missed detection and false detection in obstacle detection has become a key problem to be solved urgently. To this end, an algorithm named OSTD-YOLO (Occlusion and Small Obstacle Detection) is designed. The algorithm is built with YOLO11n as the basic model, and there are two main improvements. One is to build an Enhanced Micro-Target Feature Pyramid module EMTFP (Enhanced Micro-Target Feature Pyramid). This module relies on the PAFPN architecture and introduces SPDConv to refine the P2 feature layer, which effectively improves the feature extraction ability of small targets. At the same time, inspired by the CSP idea and OmniKernel, the CSP-OmniKernel module is developed to integrate various features extracted at different levels to make the feature information more refined. Second, the DyHead attention mechanism detection head is introduced at the output of the model, which realizes the adaptive feature enhancement of space, scale and task dimensions, and comprehensively improves the model's perception accuracy of obstacles under complex road conditions. After testing on the public dataset BDD 100 k, it is found that compared with the original YOLO11n model, the mAP50 of the OSTD-YOLO algorithm is improved by nearly 5 percentage points, and the parameter amount is 3.58 M. It shows the best balance of parameter amount and accuracy in the comparison test with other similar methods. A series of experiments fully demonstrate that the OSTD-YOLO algorithm can effectively handle obstacle detection tasks under complex road conditions, providing strong support for the practical application of autonomous driving technology.

Keywords

Complex Road Conditions, Obstacle Detection, Deep Learning, YOLO

1. 引言

随着自动驾驶技术的迅猛发展，复杂路况下的障碍物检测成为了保障行车安全的关键环节。在自动驾驶系统中，精确且高效地识别各类障碍物，如行人、车辆、交通标志与信号灯等，直接关系到自动驾驶汽车的行驶决策[1]。

随着深度学习的发展，通过构建深度神经网络结构，能够自动从大规模数据中学习到丰富且具有代表性的特征表示，极大地提升了对复杂路况下障碍物特征的刻画能力。在深度学习框架下，目标检测算法主要分为两阶段算法和单阶段算法。两阶段算法以 R-CNN [2]系列为代表，如 Faster R-CNN [3]，其首先通过区域建议网络(RPN)生成可能包含物体的候选区域，然后再对这些候选区域进行分类和精确的边界框回归[4]。这种方法虽然在检测精度上有一定优势，但由于其分阶段的处理过程，计算复杂度较高，导致检测速度相对较慢，难以直接应用于对实时性要求极高的自动驾驶场景。

而单阶段算法则在保持较高检测精度的同时，显著提升了检测速度，更契合自动驾驶系统的需求。单阶段算法直接在整个图像上进行目标检测，无需预先生成候选区域，大大减少了计算量。YOLO [5] (You Only Look Once)系列算法作为单阶段算法的典型代表，在目标检测领域取得了令人瞩目的成果。与其他 One-Stage 目标检测算法(如 SSD [6]、RetinaNet [7])相比，YOLO 算法仅通过卷积神经网络对整个图像进

行一次性预测，而其他算法需要多次滑动窗口扫描来完成目标检测。这使得 YOLO 算法在保持检测速度的同时，具备较高的检测精度和定位准确性。

近几年，在道路障碍物检测领域已有学者展开了相关的研究。例如，薛雅丽[8]等通过给 YOLOv5 算法添加小目标检测层、引入 EMA 机制与 CSL 技术，有效提升了 SAR 舰船小目标检测精度与鲁棒性。王燕妮[9]等通过优化损失函数、检测层与 C2f 模块改进 YOLOv8 算法，提升了无人机小目标检测性能。余荣威[10]等通过引入注意尺度序列融合机制、增小目标检测层、改用 RT-DETR 检测头并构建新损失函数来改进 YOLOv8 模型，提升了交通标志检测的精度与泛化力。汤伟博[11]等通过对 YOLOv8 模型添加自研模块 CSP-RFA 等，优化网络结构与损失函数，增强了无人机航拍对遮挡目标检测效果。但是，以上研究由于引入小目标检测层，导致模型复杂度以及参数量几何倍增加，算法的可部署性差，对硬件设备要求高，实时性大幅降低，实用性不高。

针对以上问题，本文提出了一种基于改进 YOLO11 的复杂路况障碍物检测方法：OSTD-YOLO 算法，提升了模型对小目标以及遮挡目标检测的性能。主要贡献如下。

首先提出了 EMTFP。基于 PAFPN 架构[12]，我们引入 SPDConv [13]从 P2 特征层提取富含小目信息的特征信息，将其提供给 P3 进行融合。然后通过 CSP-OmniKernel 模块对多层特征进行整合并反馈。这样可以有效地学习从全局到局部的特征表征，提高对遮挡目标和小目标的检测性能，同时弥补了传统添加小目标检测层带来的计算量过大、后处理更加耗时等问题。

其次在模型的输出端，引入 DyHead 注意力机制检测头[14]。通过这一机制，在网络传播进程中能够自动精准地聚焦于关键的特征区域，从而显著提升检测头的空间感知效能以及特征表达的丰富度与精准度，为目标检测任务的准确性与高效性提供了强有力的支持与保障，使得模型在面对复杂多变的检测场景时，能够更敏锐地捕捉到目标物体的特征信息，进而输出更为精准可靠的检测结果。

2. 复杂路况下的障碍物的检测方法：OSTD-YOLO

YOLO11 [15]是由 Ultralytics 团队在 2024 年 9 月 30 日发布，在之前 YOLO 版本取得的显著进步基础上，YOLO11 在架构和训练方法上进行了重大改进，使其成为各种计算机视觉任务中的通用选择。依照精度和模型的规模分为 YOLO11n、YOLO11s、YOLO11m、YOLO11l 和 YOLO11x，如表 1 所示。综合评估后本文选择在 YOLO11n 的基础上进行改进。

Table 1. Comparison of each model of YOLO11 series
表 1. YOLO11 系列各模型对比

Model	Size (pixels)	Map^{val} 50~95	Speed CPU ONNX (ms)	Speed T4 TensorRT10 (ms)	Params (M)	Flops (B)
YOLO11n	640	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11l	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

2.1. YOLO11 网络模型

YOLO11 模型由 Input、Backbone、Neck 和 Head 四个部分组成，结构如图 1 所示。YOLO11 较之前版本的主要更新在于：相较于 YOLOv8 [16]模型，YOLO11 作出了多方面的架构优化。其把 C2F 模块替

换为 C3K2 模块，这一改变有效增强了特征提取能力，能够更好地捕捉图像中的关键信息。并且，在 SPPF 模块之后新增了 C2PSA 模块，借助该模块结合通道与空间信息并协同多头注意力机制的特性，显著提升了对物体的感知精度，尤其在复杂场景以及应对小目标和部分遮挡物体时表现更为出色。此外，YOLO11 还引入了 YOLOv10 的 head 设计思想，运用深度可分离卷积削减了冗余计算量，从而大幅提升了模型的整体运算效率，使得 YOLO11 在目标检测任务中能够更快速且精准地完成任务。

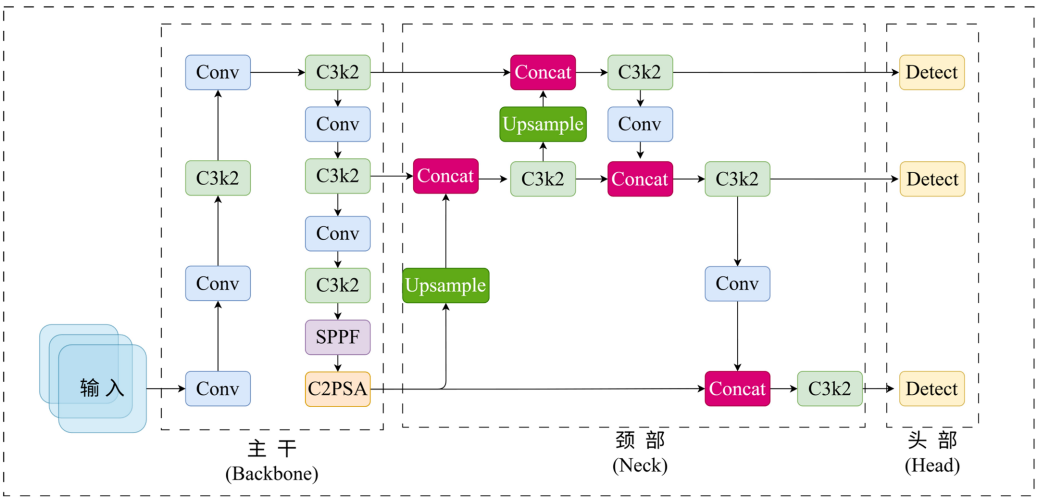

Figure 1. YOLO11 model structure
图 1. YOLO11 模型结构

2.2. OSTD-YOLO 的网络整体架构

为解决复杂场景下目标检测难题，尤其是对遮挡目标和小目标的检测挑战，我们提出了 OSTD-YOLO 算法。该算法的核心改进体现在两个关键模块。

首先是增强小目标特征金字塔模块 EMTFP。它基于 PAFPN 架构，引入 SPDConv 对 P2 特征层进行处理。SPDConv 能够有效挖掘 P2 特征层中与小目标相关的信息，增强小目标的特征表达。处理后的 P2 特征被提供给 P3 进行融合，使小目标特征在不同尺度间更好地传递和整合。之后，受 CSP 思想[17]和 OmniKernel [18]启发，设计了 CSP-OmniKernel 模块。CSP 思想合理分组特征通道，减少了特征冗余，提高计算效率；OmniKernel 的多分支结构能有效学习从全局到局部的特征表征，提升小目标检测性能，弥补了传统添加小目标检测层带来的计算量大、后处理耗时等问题。

其次，在模型输出端引入 DyHead 注意力机制检测头。在网络传播过程中，这一机制能自动精准聚焦关键特征区域，显著提升检测头的空间感知效能，使其更敏锐地感知目标物体在空间中的位置和特征。同时，它增强了特征表达的丰富度与精准度，让模型在复杂多变的检测场景中，能更好地捕捉目标物体的各种特征信息，输出更精准可靠的检测结果。

基于上述改进，我们提出了 OSTD-YOLO 算法，其结构图如图 2 所示。后续小节将对各个模块进行详细介绍。

2.3. 引入 SPDConv 模块

YOLO 算法基于深度卷积神经网络进行目标检测。在卷积神经网络提取特征信息时，通过跨步卷积和池化层操作进行下采样，虽然可以减轻计算负担，但会丢失许多细粒度信息，尤其是在处理遮挡目标以及小目标时，下采样操作导致无法得到充分的特征信息来学习。为此，引入 SPDConv 模块，该模块是

一种专门为低分辨率图像和小目标物体设计的无损下采样方法，由一个 Space-to-Depth (SPD)层和一个非跨度卷积层组成，SPD 层的作用是将输入特征图的空间维度映射到通道维度来增强特征表示，同时保证信道中的信息不会丢失。为了降低在 SPD 层中存在过度采样的可能，加入 Conv 层执行标准的卷积操作，将 SPD 层处理后的特征进行深度挖掘，增强模型对目标特征的学习能力，使得模型能够更好地捕捉目标的特征信息。如图 3 所示，是 scale = 2 时 SPDConv 对输入特征图的处理过程。

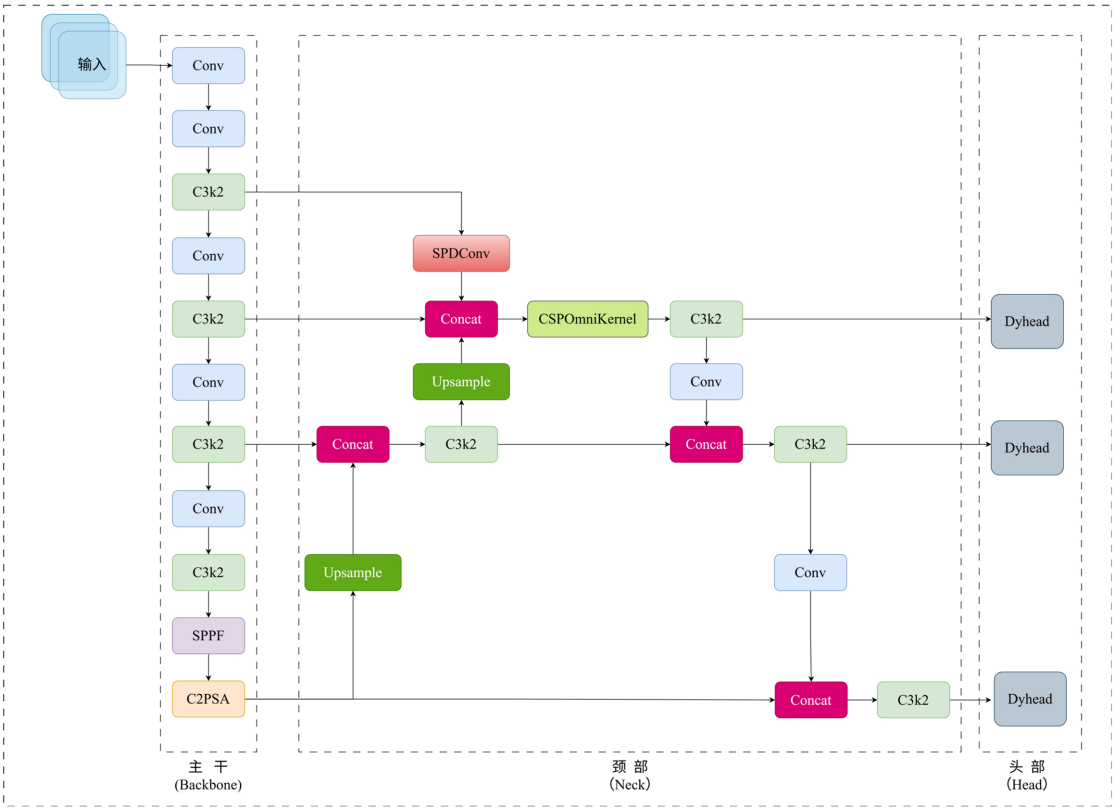

Figure 2. OSTD-YOLO model structure
图 2. OSTD-YOLO 模型结构

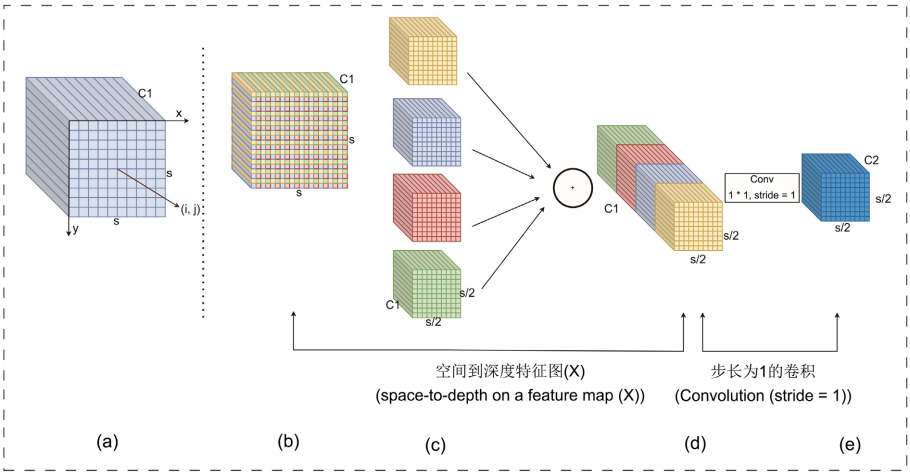

Figure 3. Schematic SPD-Conv layer scale of 2
图 3. Scale 为 2 的 SPD-Conv 层示意图

其中(a)为原始数据，(b)，(c)，(d)表示空间扩展，(e)为步长为 1 的非跨步卷积。对于尺寸大小为的原始特征图，按照比例经过 SPD 层进行裁剪操作后，可获得一系列尺寸大小均为 $S\times S\times C_1$ 的子特征图 $f_{x,y}$，如公式(1)所示为 scale 数值与子特征图的对应关系，例如当 scale = 2 时，能够得到四个子特征映射，分别为 $f_{0,0}$， $f_{1,0}$， $f_{0,1}$ 以及 $f_{1,1}$。

$$\begin{aligned}
&f_{0,0}=\mathrm{X}\left[0:S:\mathrm{scale},0:S:\mathrm{scale}\right],\\
&f_{1,0}=\mathrm{X}\left[1:S:\mathrm{scale},0:S:\mathrm{scale}\right],\\
&f_{\mathrm{scale}-1,0}=\mathrm{X}\left[\mathrm{scale}-1:S:\mathrm{scale},0:S:\mathrm{scale}\right];\\
&f_{0,1}=\mathrm{X}\left[0:S:\mathrm{scale},1:S:\mathrm{scale}\right],f_{1,1},\cdots,\\
&f_{\mathrm{scale}-1,1}=\mathrm{X}\left[\mathrm{scale}-1:S:\mathrm{scale},1:S:\mathrm{scale}\right];\\
&\cdots\\
&f_{0,\mathrm{scale}-1}=\mathrm{X}\left[0:S:\mathrm{scale},\mathrm{scale}-1:S:\mathrm{scale}\right];\cdots,\\
&f_{\mathrm{scale}-1,\mathrm{scale}-1}=\mathrm{X}\left[\mathrm{scale}-1:S:\mathrm{scale},\mathrm{scale}-1:S:\mathrm{scale}\right]
\end{aligned}\tag{1}$$

之后把所有子特征图沿着通道维度进行拼接，从而得到空间维度降低 2 倍、通道维数增加 4 倍的特征图 $\mathrm{X}'\left(\frac{S}{\mathrm{scale}},\frac{S}{\mathrm{scale}},\mathrm{scale}^2C_1\right)$ 即 $\mathrm{X}'\left(\frac{S}{2},\frac{S}{2},4C_1\right)$。在 SPD 层之后，我们添加了一个非步进(即 stride = 1)的卷积层，该层具有 C_2 个过滤器，其中 $C_2<\mathrm{scale}^2C_1$，最终输出特征图 $\mathrm{X}''\left(\frac{S}{\mathrm{scale}},\frac{S}{\mathrm{scale}},C_2\right)$。

2.4. 构建 CSP-OmniKernel 模块

我们受到 CSP 思想[17]和 AAAI2024 的 OmniKernel [18]启发，设计了 CSP-OmniKernel 模块用于特征整合。在我们的算法中，CSP-OmniKernel 模块进一步提升了模型对小目标的检测性能。通过 CSP 思想减少特征冗余和 OmniKernel 的多分支特征学习，模型能够更好地适应小目标的特点，如特征微弱、易受背景干扰等。这种适应性使得模型在面对小目标时能够更准确地识别和定位，提高了小目标检测的召回率和精度。如图 4 是 CSP-OmniKernel 模块的结构图。

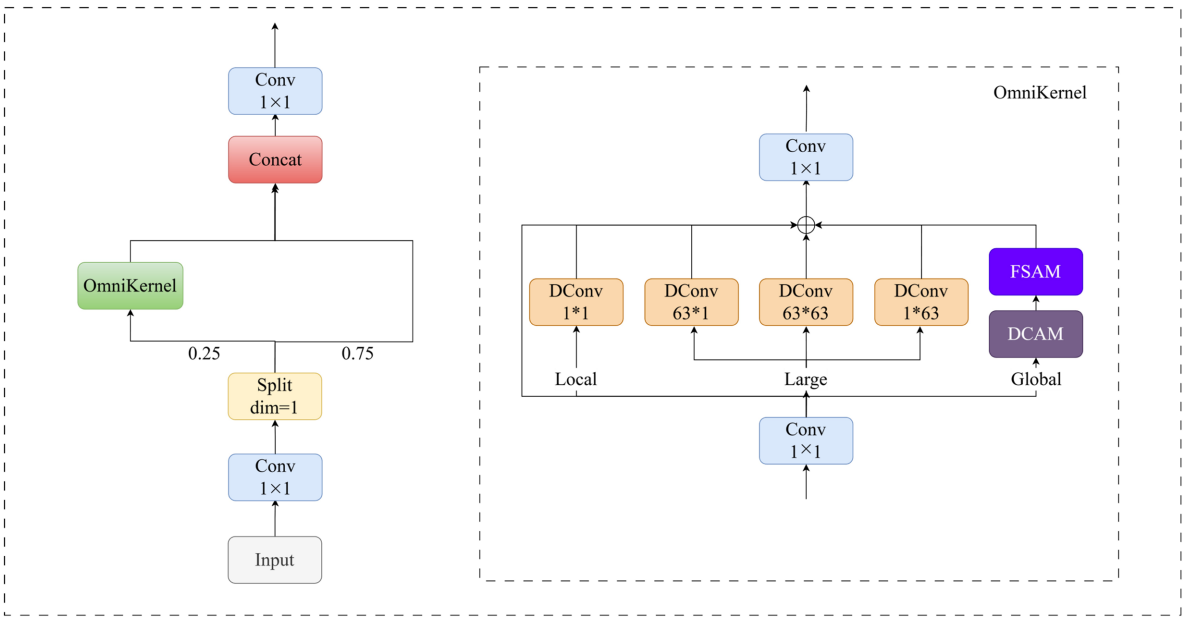

Figure 4. CSP-OmniKernel structure diagram
图 4. CSP-OmniKernel 结构图

CSP 思想在 CSP-OmniKernel 模块中通过合理分组特征通道来优化特征整合过程。它将特征通道划分为不同的组，一组保持原始特征，另一组经过特定的变换操作。这种分组方式有效减少了特征冗余，避免了不必要的信息重复计算，从而提高了计算效率。在算法中，大量的特征数据需要高效处理，CSP 思想通过这种方式使得特征在网络传递过程中更加精简和有效，为后续的检测任务提供了更优质的特征基础。

OmniKernel 设计了全局、大、局部分支结构，能够从不同的尺度学习特征表征。全局分支可以捕捉到图像的整体语义信息，为小目标的定位提供了宏观的上下文指引；大分支使用大卷积核提升较大尺度目标的特征提取和关联分析的能力；局部分支则专注于小目标的局部细节特征挖掘，能够精准地捕捉小目标的细微特征。三个分支协同工作，能够有效地学习从全局到局部的特征表征，使得模型对小目标的特征理解更加全面和深入。

2.5. 引入携带注意力机制的目标检测头 DyHead

在目标检测任务中，检测头的性能对于准确识别和定位目标起着关键作用。为了提升模型对小目标的检测能力，本研究引入了携带注意力机制的目标检测头 DyHead，通过尺度感知(Level-wise)、空间感知(Spatial-wise)和任务感知(Channel-wise)三个维度的注意力机制实现对输出结果的增强，使模型能够更好地聚焦于关键特征，从而提高检测精度。如图 5 所示为 DyHead 的网络结构图。

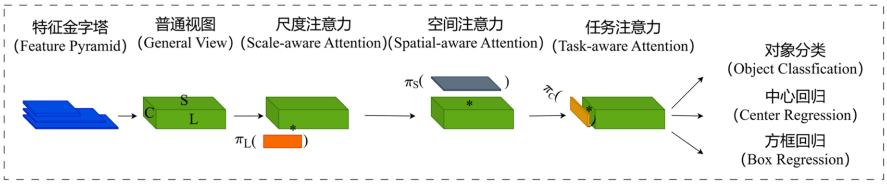

Figure 5. DyHead network structure diagram
图 5. DyHead 网络结构图

在一般视图中，L、S、C 分别代表不同尺度(不同 layer 或 stage)的特征图、空间位置信息($S=H\times W$)以及通道信息。π_L 为尺度感知注意力，π_S 是空间注意力，π_C 则是通道注意力。DyHead 是这三种注意力的叠加组合，需注意的是，这三种注意力的堆叠仅仅是一个 block，而实际的 head 是由多个这样的块叠加而成的。如图 6 所示，是 DyHead 的模块结构图。

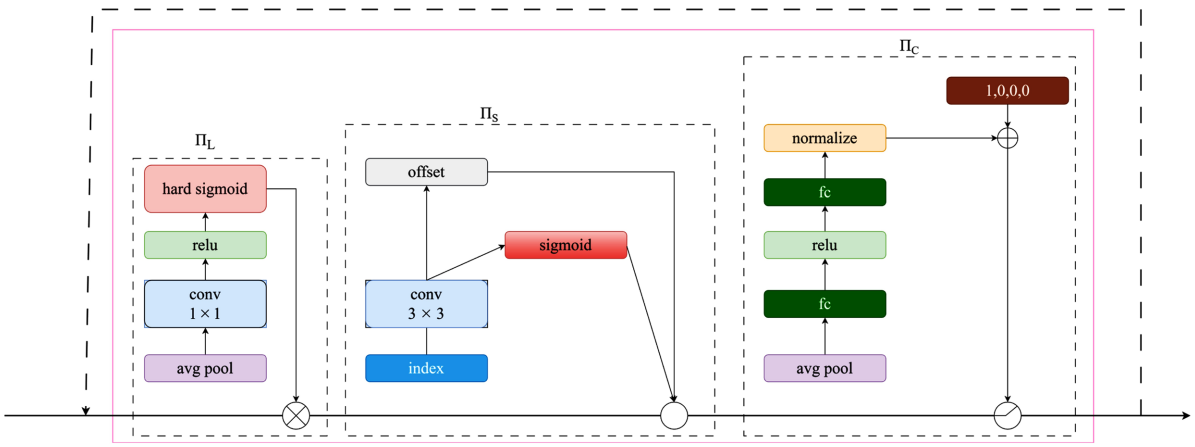

Figure 6. DyHead module structure diagram
图 6. DyHead 模块结构图

特征图借助三种注意力机制模块可得到相应输出，其关系如公式(2)所示：

$$\Pi_L(F)\cdot F=\Pi_C\left(\Pi_S\left(\Pi_L(F)\cdot F\right)\cdot F\right)\cdot F, \tag{2}$$

其中，Scale-aware 注意力 π_L 的结构包含全局池化、1 × 1 卷积、ReLU 激活函数以及 hard sigmoid 激活函数。具体而言，Adaptiv-eAvgPool2d 在 $H\times W$ 上实施全局池化，从而获取特征图的最大值，随后通过卷积将通道整合起来，最终形成 L 个数，公式为：

$$\pi_L(F)\cdot F=\sigma\left(f\left(\frac{1}{SC}\sum{}_{S,C}F\right)\right)\cdot F, \tag{3}$$

这里的 f 表示一个 1 × 1 卷积层的线性函数，hard sigmoid 公式为：

$$\sigma(x)=\max\left(0,\min\left(1,\frac{X+1}{2}\right)\right), \tag{4}$$

空间注意力 π_S 运用了可行变卷积 v2，其公式为：

$$\pi_S(F)\cdot F=\frac{1}{L}\sum_{l=1}^{L}\sum_{k=1}^{K}\omega_{l,k}\cdot F\left(l;p_k+\Delta p_k;c\right)\cdot\Delta m_k, \tag{5}$$

其中 K 为稀疏采样位置数量，$p_k+\Delta p_k$ 是经学习后偏移移动的位置，Δp_k 由可形变卷积 v2 学习获得，且集中于问题区域，Δm_k 表示在位置 Δp_k 学习的重要性。通道注意力 π_C 则采用两层全连接神经网络对通道进行建模，能够借助开关控制超参数是否学习激活的阈值，以此自动选择打开或者关闭通道，以契合不同任务需求，公式如下：

$$\pi_C(F)\cdot F=\max\left(\alpha^1(F)\cdot F_C+\beta^1(F),\alpha^2(F)\cdot F_C+\beta^2(F)\right), \tag{6}$$

这里的 max( )是超参数学习激活 Channel 的阈值，F_C 为在第 C 个通道上切分的特征。传统检测头仅从单一尺度进行检测，而 DyHead 检测头通过对缺陷目标开展多尺度处理，有效联系上下文，既能留意全局信息，又能聚焦局部重点特征信息，成功解决了小缺陷本身与下采样操作所引发的特征信息丢失问题。

3. 实验结果与分析

3.1. 数据集准备

本文采用 BDD 100 k 数据集[19]作为作为算法训练以及测试的数据集，BDD 100 k 是一个大型、多样化的自动驾驶数据集，由加州大学伯克利分校的研究团队创建。它包含约 10 万条高质量的视频片段，每段时长约 10 秒，覆盖超过 160 小时的真实世界驾驶场景，还有 10,000 个手动标注的图像。这些数据在全球多个城市、多种天气条件(如晴天、雨天、雪天等)和不同时间(白天、黄昏、夜晚)下收集，涵盖丰富的道路、交通标志和行人等元素。其优势在于数据的多样性和高质量，视频捕获了各种复杂环境，严格的手动标注确保准确性，是训练机器学习算法的理想资源。

BDD 100 k 数据集的初始标签文件，其存储格式为.JSON，而后续操作要求将其转换为 YOLO 标注数据格式。如图 7 所示，此图清晰地呈现了标签文件格式的转换示意，图中左侧部分为原始 BDD 100 k 的.JSON 标注格式相关信息，右侧对应的则是转换完成后的 YOLO 格式标注信息。在数据预处理的关键进程中，会把转换后的数据样本进一步细分为训练集与验证集。具体而言，训练集内包含有 70,000 张图像，验证集则收纳了 10,000 张图像。由于原始的 BDD 100 k 测试集本身并不涵盖标注信息，故而在整个数据处理的流程体系里，该测试集不会被纳入使用范畴。

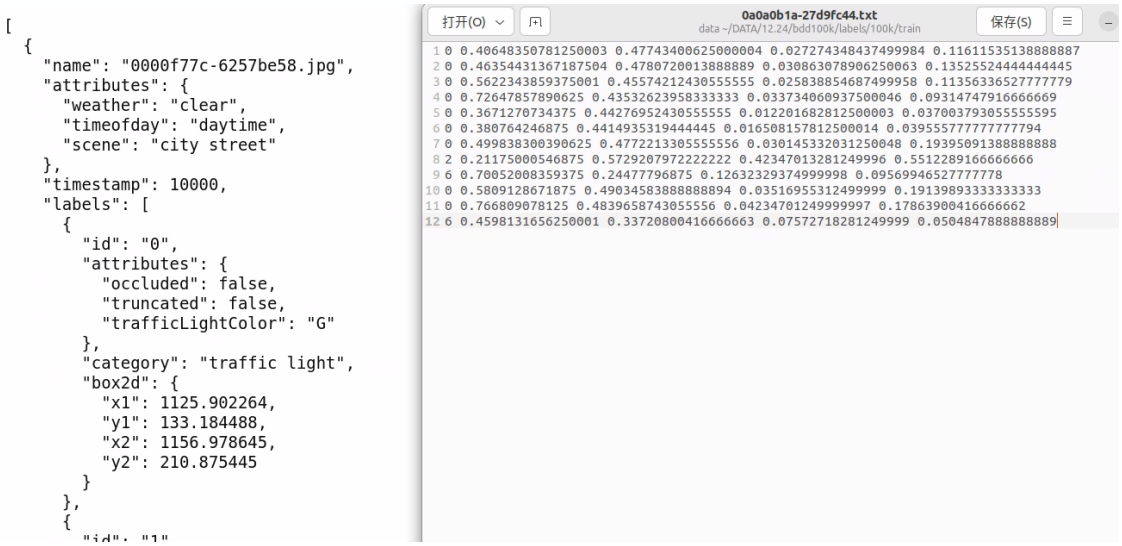

Figure 7. Label format conversion
图 7. 标签格式转化

3.2. 实验环境及评价指标

本实验基于 Ubuntu 22.04 操作系统开展，配备了两块 NVIDIA GeForce 3090 24G 的 GPU。实验环境搭建方面，采用 Python 3.8 版本，并结合 PyTorch 2.0.0 以及 cuda 11.8。在训练过程中，选用 AdamW 优化器，同时，各实验超参数依据表 2 所示进行设置。

Table 2. Model training parameter settings
表 2. 模型训练参数设置

Parameters	Settings
Epochs	300.0
batch size	64.0
Workers	8.0
image size	640.0
Lr0	0.001
patience	10.0

本文我们选用精确率(Precision, P)、召回率(Recall, R)、平均精度值(Map)、每秒帧数(Frames Per Second, FPS)以及每秒十亿次浮点运算(Giga Floating-Point Operations Per Second, Gflops)作为评价基准，对优化后的算法在检测性能方面展开全方位评估。精确率精准体现了模型准确分类的实力，具体而言，是在所有已识别的结果里，正确识别部分所占的比重；召回率则着重凸显了模型全面锁定目标的效能，也就是正确识别出的目标数量与测试集中目标总数的比例关系。平均精度值包含 mAP_{50} 与 $mAP_{50:95}$ 这两个重要指标，其中，mAP_{50} 是指在交并比(IoU)阈值设定为 0.50 时，所有目标类别平均的检测精度；$mAP_{50:95}$ 表示在以 0.05 为步长，计算 IoU 阈值从 0.50 逐步递增至 0.95 的全部 10 个不同 IoU 阈值下的检测精度，并求取其平均值，这能够充分反映模型在不同 IoU 要求情境下的整体性能表现。此外，GFLOPs 表示模型复杂程度的计算量，表示在执行模型时，所涉及的浮点运算的数量，这个值越大代表对硬件资源的要求越高，单位为 G。

3.3. 消融实验

为了验证 EMTFP 以及 DyHead 对 yolo11 算法改进的有效性，进行消融实验，如表 3 所示，从实验 B 可以看出，通过自研方法 EMTFP，模型在各方面能力上均有提升，精确率增加 2.8 个百分点，召回率增加 3.7 个百分点，mAP_{50} 增加 3.4 个百分点，$mAP_{50:95}$ 增加 2.2 个百分点，参数量仅增加 0.48 M，实验结果充分表明了自研方法的有效性；实验 C 是将基线算法的检测头更换为 DyHead，算法的整体性能也是得到了提升，其中精确率提高 1.9 个百分点，召回率提高 1.8 个百分点，mAP_{50} 提高 1.8 个百分点，$mAP_{50:95}$ 提高 1.2 个百分点，虽然相比 EMTFP 对基线算法的性能提升略低，但 GFLOPs 仅仅增加 1.2G；实验 D 是将 EMTFP 以及 DyHead 同时添加至基线算法，实验结果表明，算法的整体性能提高，在仅增长参数量 1M 的情况下，精确率提高了 3.1 个百分点，召回率提高 4 个百分点，mAP_{50} 提高了近 5 个百分点，$mAP_{50:95}$ 提高了 3.1 个百分点，能够更好的完成复杂路况下对障碍物的检测任务。

Table 3. Analysis of ablation experimental results of different improved point combinations
表 3. 不同改进点组合的消融实验结果分析

Experiment Number	EMTFP	DyHead	P/%	R/%	mAP_{50}/%	$mAP_{50:95}$/%	Params/M	GFLOPs/G
A	-	-	67.5	40.9	45.2	25.6	2.58	6.3
B	√	-	70.3	44.1	48.6	27.8	3.06	12.3
C	-	√	69.4	42.7	47	26.8	3.10	7.5
D	√	√	70.6	44.9	50	28.7	3.58	13.4

3.4. 对比实验

为充分验证本研究算法的卓越性能，我们开展了一系列严谨的训练与测试，并将其与 YOLO 系列其他算法进行深入对比。在此次对比中，选取的参照算法不仅有广泛应用的 YOLO “n” 系列，还特别纳入了作为性能标杆的 YOLOv5s 以及新兴的 Yolo11s。评价指标则聚焦于 mAP (平均精度均值)与 Params (参数量)，以此全面衡量各算法模型在检测精度与参数量方面的表现。测试结果清晰地呈现于表 4 中。数据表明，与 YOLO “n” 系列算法相比，本研究算法在平均精度上展现出显著优势。尤为突出的是，在参数量不足 YOLOv5s 一半的情况下，本研究算法在各项性能指标上均成功实现超越。即使在参数量仅为 Yolo11s 三分之一的严苛条件下，本研究算法依然达到了与 Yolo11s 相当的性能水平。另外本算法的 FPS 达 211.7 f/s 综上所述，本研究算法在确保低参数量以及实时性的基础上，显著提升了检测精度，在复杂路况下对障碍物的检测任务场景中展现出极高的实用性与应用价值。

Table 4. Detection accuracy and parameter quantities of different algorithms on BDD 100 k dataset
表 4. 不同算法在 BDD 100 k 数据集上的检测精度与参数量

Models	mAP_{50}/%	$mAP_{50:95}$/%	Params/M	GFLOPs/G
YOLOv5s	49.5	28.5	7.81	18.8
YOLOv8n	45.1	25.7	2.69	6.8
YOLOv10n	45.5	25.6	2.27	6.5
YOLO11n	45.2	25.6	2.58	6.3
YOLO11s	50.5	29.8	9.42	21.3
Ours	50.0	29.0	3.58	13.4

3.5. 可视化实验

为切实检验本研究算法于实际场景所展现出的效能，我们于 BDD 100 k 的测试集中精心遴选了涵盖遮挡、模糊、夜间、密集聚集等多种复杂场景的图片，以此展开全面且深入的测试工作，测试结果直观呈现于图 8 之中。在测试进程中，我们详尽统计了场景内所有目标以及被检测目标的具体数量，并充分检查了检测框的精准程度。如图 8 所示，第一行结果显示 YOLO11n 相对改进后的算法 OSTD-YOLO 漏检了一辆卡车，一个交通标识，并且将一个交通标识误检为两个；第二行结果显示 YOLO11n 相对改进后的算法 OSTD-YOLO 漏检了右侧的摩托车以及骑行摩托车的人；第二行结果显示 YOLO11n 相对改进后的算法 OSTD-YOLO 漏检了交通信号灯并且将路右侧的绿植误检为车辆。经严谨分析比对后发现，相较于传统的 YOLO11n 算法，本研究算法在误检率与漏检率这两项关键指标上，均取得了极为显著的优化提升。尤其是在小目标的识别任务中，本研究算法更是脱颖而出，其表现堪称卓越，精准度与稳定性远超预期。这一系列结果强有力地表明，本研究算法在应对复杂多变的实际场景，以及妥善处理诸如目标遮挡这类棘手问题时，展现出了更为强劲的鲁棒性，同时在实际应用中的实用性也得到了充分验证，能够为各类现实场景下的目标检测任务提供更为可靠、高效的解决方案。

a.原始图像　　b.YOLO11n检测结果图　　c.OSTD-YOLO检测结果图

Figure 8. Visual comparison of effects before and after model improvement
图 8. 模型改进前后效果可视化对比

4. 结论

在自动驾驶技术蓬勃发展的当下，复杂路况下障碍物检测漏检误检的问题亟需解决。本研究针对此问题，提出了 OSTD-YOLO 算法，有效提升了模型在复杂路况下对小目标及遮挡目标的检测性能。通过对传统检测方法和基于深度学习检测方法的分析，明确了 YOLO 系列算法在目标检测领域，尤其是自动驾驶场景中的优势。本研究在 YOLO11n 的基础上进行创新改进，提出了一系列有效的策略。首先在网络

架构改进方面，基于 PAFPN 提出 EMTFP 模块，引入 SPDConv 处理 P2 特征层，增强小目标特征表达，同时避免了传统添加 P2 检测层带来的计算量过大等问题。然后运用 CSP 思想和 OmniKernel 改进得到 CSP-OmniKernel 模块，进一步优化特征整合，提升小目标检测性能。最后，在模型输出端引入 DyHead 注意力机制检测头，从多维度注意力机制增强检测头性能，提高检测精度。实验结果充分验证了算法的有效性。消融实验表明，EMTFP 和 DyHead 有效且效果符合设计，二者结合可使算法在参数量仅增加 1M 的情况下，精确率提高 3.1 个百分点，召回率提高 4 个百分点，$mAP_{50:95}$ 提高近 5 个百分点。对比实验显示，与 YOLO 系列其他算法相比，本研究算法在平均精度上优势明显，且 FPS 达 211.7 f/s，满足实时性要求。可视化实验证明，本研究算法在复杂场景下误检率和漏检率显著降低，在小目标识别方面表现卓越，实用性更强。

综上所述，OSTD-YOLO 算法在复杂路况障碍物检测任务中展现出高实用性和应用价值，为自动驾驶技术的发展提供了有力支持。未来研究可进一步探索优化算法，以适应更多复杂场景和任务需求。

参考文献

[1] 赵洋, 王潇, 蔡柠泽, 等. 自动驾驶目标检测不确定性估计方法综述[J]. 汽车工程学报, 2024, 14(5): 760-771.

[2] Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014) Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. 2014 *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 580-587. https://doi.org/10.1109/cvpr.2014.81

[3] Ren, S., He, K., Girshick, R. and Sun, J. (2017) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**, 1137-1149. https://doi.org/10.1109/tpami.2016.2577031

[4] 李晓晖, 夏芹, 张强. 基于路侧视觉感知的交通目标检测及跟踪方法研究[J/OL]. 汽车工程学报, 2024: 1-10. http://kns.cnki.net/kcms/detail/50.1206.U.20240707.1822.002.html, 2025-04-29.

[5] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You Only Look Once: Unified, Real-Time Object Detection. 2016 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Las Vegas, 27-30 June 2016, 779-788. https://doi.org/10.1109/cvpr.2016.91

[6] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., *et al.* (2016) SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N. and Welling, M., Eds., *Computer Vision—ECCV* 2016, Springer, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

[7] Lin, T., Goyal, P., Girshick, R., He, K. and Dollar, P. (2017) Focal Loss for Dense Object Detection. 2017 *IEEE International Conference on Computer Vision* (*ICCV*), Venice, 22-29 October 2017, 2999-3007. https://doi.org/10.1109/iccv.2017.324

[8] 薛雅丽, 贺怡铭, 崔闪, 等. 基于改进 YOLOv5 的 SAR 图像有向舰船目标检测算法[J]. 浙江大学学报(工学版), 2025, 59(2): 261-268.

[9] 王燕妮, 张婧菲. 改进 YOLOv8 的无人机小目标检测算法[J/OL]. 探测与控制学报, 2024: 1-10. http://kns.cnki.net/kcms/detail/61.1316.TJ.20241212.1003.005.html, 2025-04-29.

[10] 余荣威, 张逸轩, 曹书明, 等. 基于改进 YOLOv8 模型的交通标志检测方法[J/OL]. 武汉大学学报(理学版), 2024: 1-10. https://doi.org/10.14188/j.1671-8836.2024.0077, 2025-04-29.

[11] 汤伟博, 方强, 李沛根, 等. 基于 RSD-YOLO 的无人机航拍图像的小目标检测[J/OL]. 计算机工程, 2024: 1-15. https://doi.org/10.19678/j.issn.1000-3428.0070151, 2025-04-29.

[12] Liu, S., Qi, L., Qin, H., Shi, J. and Jia, J. (2018) Path Aggregation Network for Instance Segmentation. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 8759-8768. https://doi.org/10.1109/cvpr.2018.00913

[13] Sunkara, R. and Luo, T. (2023) No More Strided Convolutions or Pooling: A New CNN Building Block for Low-Resolution Images and Small Objects. In: Amini, M.R., Canu, S., Fischer, A., Guns, T., Kralj Novak, P. and Tsoumakas, G., Eds., *Machine Learning and Knowledge Discovery in Databases*, Springer, 443-459. https://doi.org/10.1007/978-3-031-26409-2_27

[14] Dai, X., Chen, Y., Xiao, B., Chen, D., Liu, M., Yuan, L., *et al.* (2021) Dynamic Head: Unifying Object Detection Heads with Attentions. 2021 *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Nashville, 20-25 June 2021, 7369-7378. https://doi.org/10.1109/cvpr46437.2021.00729

[15] Khanam, R. and Hussain, M. (2024) YOLOv11: An Overview of the Key Architectural Enhancements. arXiv: 2410.17725.

[16] Varghese, R. and Sambath, M. (2024) YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness. 2024 *International Conference on Advances in Data Engineering and Intelligent Computing Systems* (*ADICS*), Chennai, 18-19 April 2024, 1-6. https://doi.org/10.1109/adics58448.2024.10533619

[17] Wang, C., Mark Liao, H., Wu, Y., Chen, P., Hsieh, J. and Yeh, I. (2020) CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. 2020 *IEEE*/*CVF Conference on Computer Vision and Pattern Recognition Workshops* (*CVPRW*), Seattle, 14-19 June 2020, 1571-1580. https://doi.org/10.1109/cvprw50498.2020.00203

[18] Cui, Y., Ren, W. and Knoll, A. (2024) Omni-Kernel Network for Image Restoration. *Proceedings of the AAAI Conference on Artificial Intelligence*, **38**, 1426-1434. https://doi.org/10.1609/aaai.v38i2.27907

[19] Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., *et al.* (2020) BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. 2020 *IEEE*/*CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Seattle, 13-19 June 2020, 2633-2642. https://doi.org/10.1109/cvpr42600.2020.00271