SST-YOLO: An Improved Autonomous Driving Object Detection Algorithm Based on YOLOv8

Qinsheng Du , Ningbo Zhang , Wenqing Bi , Ruidi Zhu , Yuhan Liu , Chao Shen , Shiyan Zhang , Jian Zhao *

Posted Date: 25 February 2026

doi: 10.20944/preprints202602.1454.v1

Keywords: YOLOv8; autonomous driving; Sobel convolution; SOAPN; TADAHead

 doi:10.20944/preprints202602.1454.v1

Article

SST-YOLO:An Improved Autonomous Driving Object Detection Algorithm Based on YOLOv8

Qinsheng Du [1,2,3], Ningbo Zhang [1,3],,Wenqing Bi [1,3], Ruidi Zhu [1,3], Yuhan Liu [1,3], Chao Shen [1,2], Shiyan Zhang [1,3] and Jian Zhao [1,2,3,*]

[1] College of Computer Science and Technology, Changchun University, Satellite Road, Changchun, 130022, Jilin Province, China.

[2] Jilin Rehabilitation Equipment and Technology Engineering Research Center for the Disabled, Changchun University, Satellite Road, Changchun, 130022, Jilin Province, China.

[3] Ministry of Education Key Laboratory of Intelligent Rehabilitation and Barrier-Free Access for the Disabled, Changchun University,Satellite Road, ChangChun, 130022, Jilin Province, China.

* Correspondence: zhaojian@ccu.edu.cn

Abstract

As autonomous driving technology progresses, efficient and accurate object detectors are able to detect pedestrians, vehicles, road signs, and obstacles in real time, thereby enhancing driving safety and serving as a part of autonomous driving. However, the performance of such object detectors is limited and cannot be leveraged to satisfy a modern autonomous driving system. To address this issue, we develop an object detection network for autonomous driving scenarios, SST-YOLO, which is based on YOLOv8. Specifically, we propose a Sobel convolution & convolution (SCC) to enhance the backbone network of YOLOv8 and perform more full feature extraction. In addition, we replace the original path aggregation feature pyramid network (PAFPN) with a small object augmentation pyramid network (SOAPN) to solve the small object detection problem. For regression accuracy and classification robustness, and thereby to increase the performance in a complex driving scenario, we use a Task-Adaptive Decomposition & Alignment Head (TADAHead) to replace the old YOLOv8 detection head. Experiments on the public autonomous driving dataset KITTI also show that our proposed method outperforms the baseline YOLOv8 model. Compared with the baseline results, the detection accuracy ranges from 65.1% to 68.2%, which indicates that the proposed SST-YOLO network can achieve object detection for autonomous cars.

Keywords: YOLOv8; autonomous driving; Sobel convolution; SOAPN; TADAHead

1. Introduction

With the development of artificial intelligence and intelligent transportation technologies, autonomous driving [1] has become one of the most important applications of computer vision [2]. Object detection [3] is a core component of autonomous driving perception. It detects and localizes traffic drivers and the environment around the vehicle (e.g., vehicle, pedestrians, cyclists, traffic signs), and the performance of object detection algorithms directly determines the safety, reliability and decision-making performance of autonomous vehicles. Object detection has the dual role of detecting both the locations and classes of targets in images or point clouds [4]. Since the dynamics of real-world traffic environments (e.g., different object sizes, occlusions, illuminations and backgrounds) are very complex, the accuracy and real-time performance of object detection technology are correlated with vehicle safety and reliability. The accuracy and performance of autonomous driving object detection must meet real-time and robustness constraints and are subject to high algorithmic performance requirements and resistance to environmental interference. With the advancement of deep learning, object detection methods have moved from handcrafted feature-

 doi:10.20944/preprints202602.1454.v1

based methods to deep learning-based approaches and have provided technical support for autonomous driving perception systems.

In real-world road conditions, autonomous vehicles must work reliably in complex and dynamic scenes. The large size, dense scenes, occlusions, illumination effects, and bad weather make object detection very difficult. For example, small objects [5] and long-distance targets [6] are highly likely to be missed, which can be a threat to safety. Hence, improving the ability to detect objects in complicated driving environments is still a major challenge in future studies.

The emergence of deep learning [7] has revolutionized object detection. Detection methods based on CNNs [8] automatically learn hierarchical feature representations from large-scale datasets. Current deep learning-based object detection methods can be broadly divided into two types: two-stage-based detection [9] and one-stage detection [10]. Two-stage methods (e.g., R-CNN, Fast R-CNN and Faster R-CNN) achieve high detection accuracy by decoupling region proposal generation from classification and regression, but they are costly and have low-time performance. One-stage detectors such as SSD and YOLO family detectors design object detection as a single regression problem, which has advantages such as end-to-end inference [11], faster detection speed, and simpler network architectures and can be readily employed in real-time applications such as autonomous driving. With the development of deep information technology, deep learning-based object detection techniques are becoming very active in computer vision. Compared with traditional algorithms, they learn discriminative image features and can achieve enormous performance gains. Despite the state-of-the-art performance of one-step-based object detectors, they can still perform poorly in detecting small- and long-distance objects in autonomous driving scenarios. To address this issue, we propose a YOLOv8-based object detector based on Sobel Convolution & Convolution (SCC), a small object augmentation pyramid network (SOAPN) and a decomposed and task-adaptive decomposition & alignment head (TADAHead). Our contributions can be summarized as follows:

(1) An SCC module is introduced to improve the C2f modules in YOLOv8. With a Sobel operator [12] branch to enlarge edge information and combine it with traditional convolutional networks for feature extraction, multilevel information fusion can be achieved, and the model is more sensitive to fine-grained and detailed features.

(2) A SOAPN is adopted to replace the PAFPN in the neck of YOLOv8 to improve the representation and detection performance of small objects.

(3) The initial detection head of YOLOv8 is replaced with a TADAHead. When the model is run, the proposed head provides regression accuracy and classification robustness via taskwise dynamic decoupling and spatial alignment.

(4) We conduct extensive experiments on the public autonomous driving dataset KITTI and show that SST-YOLO is superior to the other YOLOv8s.

2. Related Work

2.1. Overview Of Object Detection

Object detection, which aims to find all the objects of interest in an image and the categories and locations associated with it, is one of the most challenging problems in computer vision. This is a crucial task in many applications, such as autonomous driving, video surveillance and medical image analysis. Owing to the diversity of the appearance, shape and pose of objects and due to issues such as illumination variations and occlusions during image acquisition, object detection has become one of most challenging tasks in computer analysis. Existing object detectors can be broadly classified into two categories: 2-stage detectors and 1-stage detectors. The two-stage detector automatically generates a set of selected regions and then performs classification and bounding box regression on the regions. Examples include R-CNN [13], Fast R-CNN [14] and Faster R-CNN [15]. However, the one-stage detection algorithm performs the input image directly and predicts the object categories and locations. Examples in SSD [16] and YOLO [17] are the most popular.

 doi:10.20944/preprints202602.1454.v1

2.2. Two-Stage Object Detection Algorithms

The R-CNN employs the selective search (SS) [18] algorithm to generate region proposals. The candidate regions are cropped from the original image and resized to a fixed resolution, after which convolutional neural networks are used for feature extraction and classification. Each region yields a confidence score and a bounding box offset, and nonmaximum suppression (NMS) is applied to high-confidence proposals to obtain the final detection results. The introduction of convolutional neural networks significantly improves detection accuracy. However, the large number of redundant region proposals, which is a characteristic of R-CNN, leads to excessive repetitive computations, severely limiting the detection speed and becoming a major bottleneck in the algorithm's performance.

Fast R-CNN, an improved version of R-CNN, integrates the advantages of R-CNN and SPPNet. First, it adopts a single convolutional forward pass, as in SPPNet, to avoid redundant feature extraction. Second, the traditional SVM classifier is replaced with a Softmax function for classification, and multiple parallel output layers are introduced at the end of the network to enable end-to-end multitask learning for classification and bounding box regression without requiring additional feature storage. Finally, a region of interest (RoI) pooling layer is designed to pool region proposals of different sizes on the feature map into fixed-size feature representations before being fed into the fully connected layers. As a result, the Fast R-CNN achieves a significantly faster detection speed than the R-CNN while maintaining comparable detection accuracy.

Region proposal generation has long been one of the key factors limiting the performance of deep learning–based detection algorithms until this issue was effectively addressed by Faster R-CNN, which led to a substantial improvement in detection speed. Faster R-CNN introduces a region proposal network (RPN) that formulates region proposal generation as a binary classification problem. Specifically, anchors of multiple scales and aspect ratios are generated and slid over the feature map, after which they are labeled as positive or negative according to predefined thresholds. The RPN outputs the anchor coordinates along with the objectness scores, which are subsequently used as region proposals. End-to-end training significantly improves the quality of region proposals, replacing the previously time-consuming proposal-and-detection pipeline and enabling nearly cost-free region proposal generation. Faster R-CNN is an integrated framework that organically combines the RPN with Fast R-CNN.

2.3. Single-Stage Object Detection Algorithms

Noting that the first step has been followed by the two-stage detection task "proposal first, detection later," R. Joseph et al. proposed the first version of YOLO (you only look once) in 2015 as a regression problem. YOLO is a simple image split method that involves partitioning the input image into a 7×7 grid, and for each grid cell, the algorithm determines which center of an object belongs to the grid and performs object detection. Instead of having the RPN improve the proposal efficiency with training cost, YOLO generates 49 detection regions via image splitting, with fewer regions and better detection efficiency, and fixed splitting does not introduce additional training overhead. YOLO has limited generalizability, since every grid cell predicts only one object class, and small-object center localization has a strong effect on the loss function. It does not perform well in small object detection and does not yield good generalization performance for irregular object shapes.

The single shot multibox detector (SSD) is a detector based on multiscale and multiresolution representations. Unlike YOLO, SSD performs object detection directly during convolution. By adding multiscale convolution, SSD makes predictions at different scales for small objects of different sizes, which reduces the loss from scale variation. In particular, shallow feature maps are used to detect small objects to preserve fine details, and deeper feature maps can be used for large object detection to capture more semantic information. However, it has its shortcomings. Detection at multiple scales introduces redundant computations and increases the overall computational complexity. Moreover, shallow feature maps retain more details for small objects, but they do not provide enough semantic representation for small detection.

 doi:10.20944/preprints202602.1454.v1

2.4. Yolov8 Network

The eighth version of the YOLO network, YOLOv8, is a state-of-the-art (SOTA) model that builds upon the success of previous YOLO versions by introducing new features and architectural improvements to further enhance performance and flexibility. YOLOv1 completes object detection with a single forward pass; however, it performs poorly in detecting small objects and has limited accuracy in complex scenarios. YOLOv2 improves upon YOLOv1 by introducing techniques such as batch normalization and anchor boxes, significantly enhancing model stability and detection accuracy. YOLOv3 adopts a new feature extraction backbone, Darknet-53, and employs multiscale detection, resulting in improved performance for objects of varying sizes. YOLOv5 is implemented via the PyTorch framework and achieves a favorable balance between inference speed and detection accuracy. Building on YOLOv5, YOLOv8 introduces multiple refinements and optimizations, preserving the advantages of the YOLOv5 architecture while further improving performance across diverse application scenarios.

The network architecture of YOLOv8 consists of the following three main components:

Backbone:the backbone is responsible for feature extraction and is composed of a series of convolutional and deconvolutional layers. Residual connections and bottleneck structures are employed to reduce network complexity while enhancing performance. YOLOv8 adopts the C2f module as its basic building block. Compared with the C3 module used in YOLOv5, the C2f module has fewer parameters and exhibits superior feature extraction capability.

Neck:the neck utilizes multiscale feature fusion to integrate feature maps from different stages of the backbone, thereby enhancing feature representation. Specifically, the neck of YOLOv8 includes an SPPF module, a PAA module, and two PAN modules, which jointly facilitate efficient information aggregation across multiple scales.

Head:the head is responsible for the final object detection and classification tasks. It consists of a detection head and a classification head. The detection head comprises a series of convolutional and deconvolutional layers to generate detection results, whereas the classification head employs global average pooling to perform category prediction for each feature map.

The overall architecture of YOLOv8 is illustrated in Figure 1.

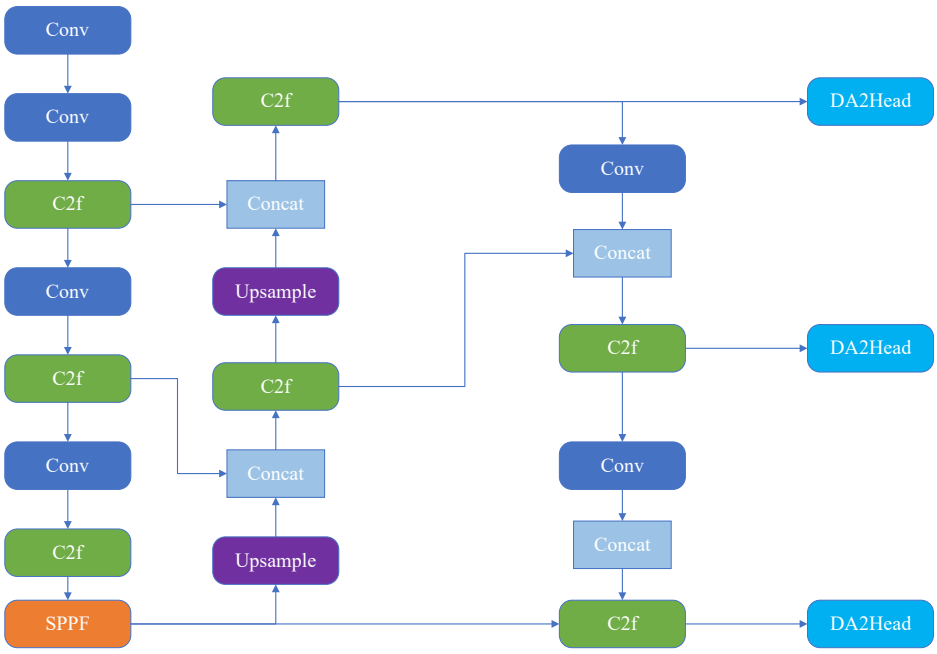

Figure 1. Structural diagram of YOLOv8.

 doi:10.20944/preprints202602.1454.v1

2.5. Feature Fusion

Feature fusion is a method that integrates feature information from different layers or branches during image processing. Key fusion tasks include channel concatenation, elementwise addition and multiplication. By incorporating complementary information from multiple feature levels, feature fusion helps the image representation of a detection model.

For scene detection tasks, Mingxing Tan et al. from Google Brain proposed an architecture called EfficientDet [19], where a weighted bidirectional feature pyramid network (BiFPN) in the neck of the network uses learnable weights to automatically reuse and fuse features from multiple input levels and performs multiple-scale feature fusion with more accurate detection accuracy.

3. Method

In the development of the autonomous driving industry, object detection systems play a crucial role. YOLOv8 is a detection framework composed of a backbone, a neck, and a detection head. Although YOLOv8 achieves high efficiency, its performance in small object detection remains suboptimal. To address this limitation, the original YOLOv8 network is improved in this work, resulting in a novel detection model named SST-YOLO, whose overall architecture is illustrated in Figure 2. The proposed model consists of several key components. In the backbone, an SCC module is introduced to enhance edge feature extraction. This modification alleviates the limitation of the original C2f module, which relies heavily on deep convolutional operations and thus has insufficient ability to capture the edge information of critical targets such as pedestrians, lane markings, and traffic signs in autonomous driving scenarios. In the neck, a small object–oriented augmentation pyramid network (SOAPN) is adopted. This structure is specifically designed for small object detection and enables more effective aggregation of multiscale features, enhancing feature representation while reducing computational complexity. For the detection head, a task-aligned decoupled adaptive head (TADAHead) is employed, which demonstrates strong performance in complex scene detection. TADAHead effectively mitigates conflicts in multitask learning and improves the spatial adaptability of feature locations through a dynamic convolution mechanism, thereby providing stronger support for the detection accuracy and robustness of the YOLOv8 detector in complex autonomous driving environments.

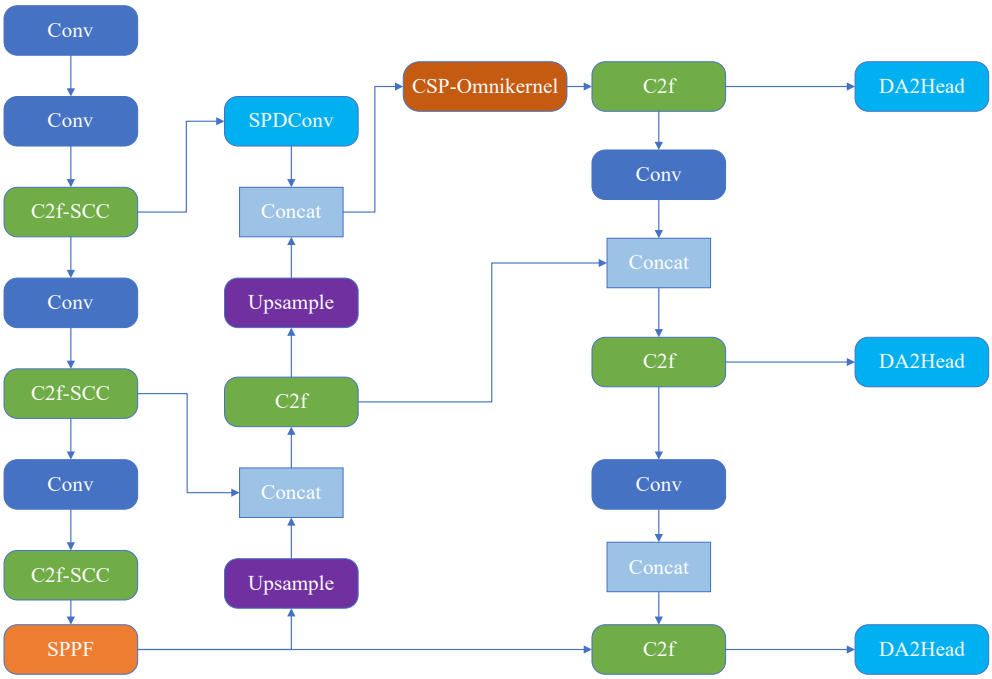

Figure 2. Structural diagram of SST-YOLO.

 doi:10.20944/preprints202602.1454.v1

3.1. SCC

In autonomous driving object detection tasks, the effectiveness of feature extraction is critical to both real-time performance and detection accuracy. Although the C2f (cross-stage partial fusion) module can effectively facilitate feature fusion, it still has several limitations. First, C2f essentially remains a conventional convolutional stacking structure and lacks explicit mechanisms for extracting image gradients and edge features. However, the most critical targets in autonomous driving scenarios—such as pedestrian contours, vehicle boundaries, lane markings, and traffic sign edges—often rely on clear edge information for accurate recognition. This limitation leads to insufficient sensitivity of C2f to small objects and weak-texture targets.

Second, C2f relies solely on standard convolutions for texture feature extraction without introducing additional structural enhancements. In real-world road environments, where occlusions, rain or fog, and low image clarity are common, local geometric structures become more crucial. Moreover, closely adjacent objects require stronger edge discrimination capabilities. The lack of structural diversity in the traditional C2f module and its insensitivity to fine-grained texture details make it difficult to meet these demands.

Finally, C2f is not well suited for small-scale object detection. In autonomous driving scenarios, many key targets—such as pedestrians, cyclists, traffic lights, and road signs—often occupy only tens of pixels. After downsampling, the convolutional operations in C2f tend to lose critical edge features, resulting in high miss-detection rates for small objects and large localization errors for distant targets.

To address these issues, this study introduces an SCC (SobelConv–Conv) module constructed via the Sobel operator, which is used to improve the traditional C2f module, forming a new C2f-SCC module. The structures of the SCC module and the C2f-SCC module are illustrated on the left and right sides of Figure 3, respectively. The proposed module is capable of extracting features from the original image while preserving rich spatial information, and it can effectively capture abrupt intensity changes to obtain critical edge information.

Conventional convolutional operations are proficient at learning spatial features but are often insufficient for explicit edge extraction. In contrast, the SCC module introduces a SobelConv branch to extract edge features explicitly. The Sobel filter is a classical edge detection operator that effectively captures sharp intensity variations in images, thereby highlighting important edge information. In addition to edge features, spatial information is equally important. Therefore, SCC incorporates an additional convolutional branch to extract features from the original image, preserving rich spatial details. Finally, features extracted from the SobelConv and standard convolution branches are concatenated, enabling the learned representation to encode rich edge information and spatial features simultaneously, thus providing a more comprehensive characterization of the image content.

 doi:10.20944/preprints202602.1454.v1

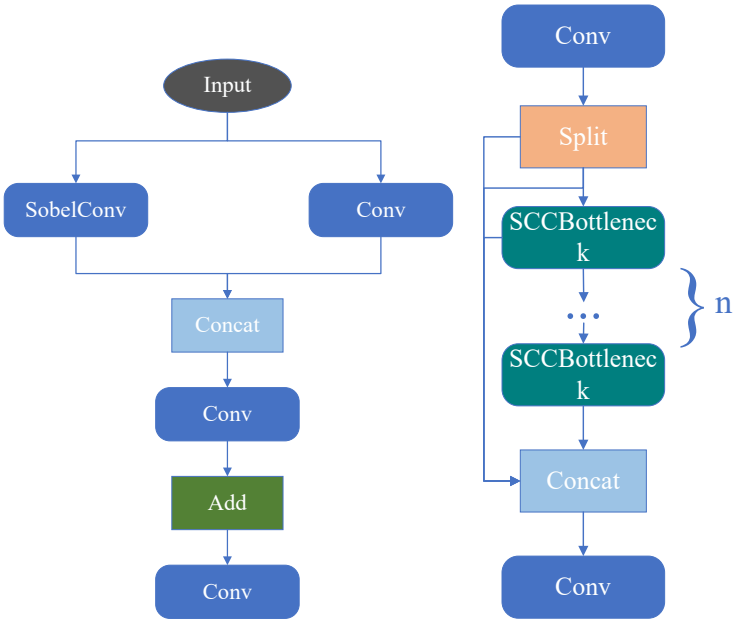

Figure 3. Structural diagram.(a)SCC;(b)C2f-SCC.

The workflow of the proposed module is described below.

Given an input feature map Xextracted from the original image, the feature map is first fed into the SobelConv branch, where the Sobel operator is applied to compute edge information in the horizontal and vertical directions. Let S_x and S_y denote the horizontal and vertical Sobel kernels, respectively. The edge feature X_{sobel} is computed as:

$$X_{\text{sobel}} = | X * S_x | + | X * S_y |,$$

where $*$ denotes the convolution operation.

Moreover, the input feature map X is processed by the Conv branch, which extracts features from the original image via standard 3×3convolution. The resulting feature is denoted as:

$$X_{\text{conv}} = f_{\text{conv}}(X),$$

where f_{conv} represents the 3×3 convolution operation.

The outputs of the SobelConv branch and the Conv branch are subsequently concatenated along the channel dimension to obtain the fused feature X_{concat}:

$$X_{\text{concat}} = [X_{\text{sobel}}, X_{\text{conv}}],$$

where $[\ \cdot\ ]$ denotes channelwise concatenation.

To integrate the fused information and reduce channel redundancy, a 1×1convolution is applied for channel compression, yielding the integrated feature X_{feature}:

$$X_{\text{feature}} = f_{1\times1}(X_{\text{concat}}),$$

where $f_{1\times1}$ denotes the 1×1 convolution operation.

Finally, another 1×1 convolution is applied after introducing a residual connection with the original input feature map X, resulting in the enhanced feature map X':

$$X' = f_{1\times1}(X_{\text{feature}} + X),$$

where the residual connection ensures information completeness and stabilizes feature learning.

By incorporating the Sobel operator, the proposed module explicitly enhances the model's ability to capture edge information. Compared with the original C2f module in YOLOv8, the improved module achieves higher detection accuracy and recall, thereby providing a solid foundation for object detection tasks in autonomous driving scenarios.

 doi:10.20944/preprints202602.1454.v1

3.2. SOAPN

A multiscale feature fusion networks, the path aggregation feature pyramid network (PAFPN) [20], which is widely adopted in the YOLO series, enables bidirectional feature interaction through top-down and bottom-up pathways. However, for diverse road object types in autonomous driving environments—such as pedestrians, cyclists, long-distance traffic signs, low-reflective obstacles at night, and nonrigid objects occluded by vehicles—the traditional PAFPN architecture exhibits several inherent structural bottlenecks, making it insufficient for meeting the robustness requirements of complex road scenes.

First, PAFPN relies heavily on a full-channel bidirectional propagation mechanism during feature fusion. High-level semantic features are progressively upsampled and transmitted to lower layers in a linear topological manner, which leads to gradual dilution of deep semantic information during reverse propagation, making it difficult to adequately recover fine-grained texture details. Second, the convolutional kernel sizes in PANet are fixed. When facing traffic targets with extreme scale variations (e.g., pedestrians at a distance of 50 m versus traffic cones within 1 m), the limited receptive field makes it difficult to balance global contour perception and local texture extraction, resulting in fragmented and blurred feature representations. Third, PAFPN emphasizes single-path feature fusion and lacks cross-scale redundant information feedback mechanisms. As a result, in dynamic and complex scenarios—such as curved lane markings, vehicle interactions, and illumination degradation under rainy or foggy conditions—the network exhibits insufficient robustness to occlusion and reflective noise.

On the basis of the above analysis, this study redesigns the feature fusion pathway of YOLOv8 and proposes the small object–oriented augmentation pyramid network (SOAPN) as a replacement for PAFPN. SOAPN enhances the bidirectional representation of global semantics and local details through heterogeneous-scale convolutions, multidimensional receptive field mixing, and lightweight cross-layer fusion. Consequently, the detection network demonstrates stronger sensitivity to small objects, improved long-distance recognition capability, and enhanced robustness to occlusion in autonomous driving scenarios.

The SOAPN architecture consists of an SPDConv [21] downsampling enhancement module and a CSP-OmniKernel feature fusion module, corresponding to the feature compression stage and the multiscale receptive field aggregation stage, respectively. Specifically, SPDConv is designed to replace conventional stride-2 convolutions by reducing semantic loss through four-dimensional reencoding of input features. Given an arbitrary input feature map

$X_{\text{in}} \in \mathbb{R}^{C\times H\times W}$,

SOAPN performs even–odd index partitioning as follows:

$$X_{\text{SPD}} = \text{Conv}([X^{0,0} \parallel X^{1,0} \parallel X^{0,1} \parallel X^{1,1}]),$$

where

$$X^{i,j} = X_{in}[:,:,i::2,j::2]$$

SPDConv first splits the spatial information into the channel dimension via a space-to-depth operation without losing the pixel information; then, a convolution with stride 1 is used to fuse and compress features to avoid loss of information due to traditional stride convolutions. This design helps local context modeling and is especially useful in small object detection.

The working principle of SPDConv is illustrated in Figure 4.

 doi:10.20944/preprints202602.1454.v1

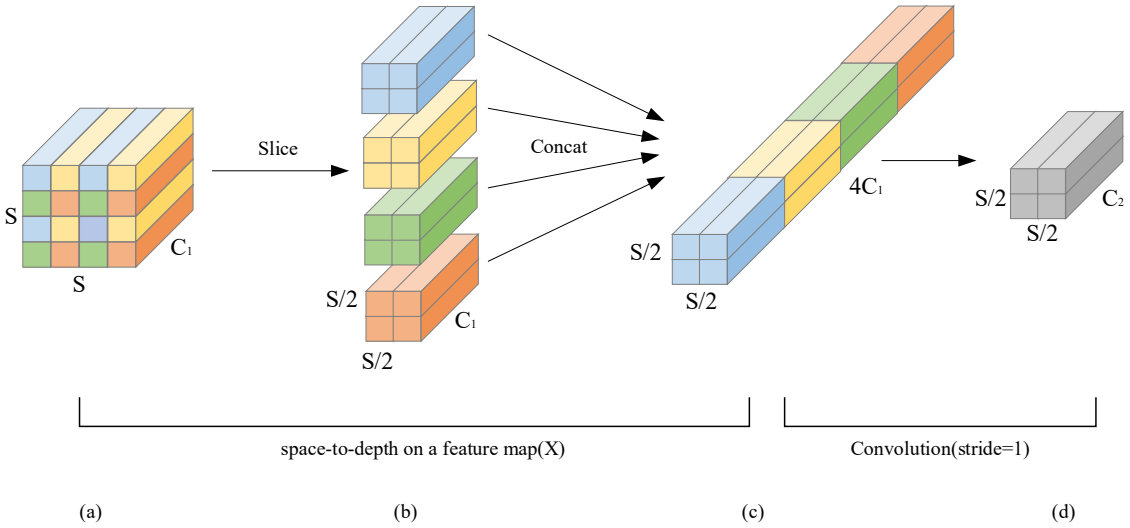

Figure 4. Schematic diagram of SPDConv.

A CSP-OmniKernel module is subsequently introduced in the multiscale feature fusion stage. This module leverages the cross-stage partial (CSP) mechanism [22] to split the input channels into two parts: one part is fed into a dilated convolution group for multiscale feature fusion, while the other part is preserved through identity mapping to retain the original texture information. This design effectively prevents the loss of structural details caused by deep convolutional operations. The process is defined as follows:

$$X_c = [X_e \parallel X_r] \text{s.t.} C_e = \lfloor C \cdot e \rfloor, C_r = C - C_e$$
$$Y = \text{Conv}_2(\ O(X_e)\ \parallel\ X_r\ )$$

where X_e and X_r denote the enhanced and residual feature subsets, respectively; C represents the number of input channels; e represents the channel split ratio; and $\parallel$ denotes channelwise concatenation.

The operator $O(\cdot)$ denotes the OmniKernel multiscale processing stream [23], which can be further expressed as:

$$O(X_e) = \sum_{k \in K} \sum_{d \in D} \text{Conv}_{(k,d)}(X_e),$$

where K represents the number of convolution kernels and D denotes the number of dilation rates. By combining heterogeneous kernels with dilation factors, OmniKernel is able to expand the receptive field and capture context information at multiple scales.

The OmniKernel module adopts a multibranch architecture to model receptive fields at multiple scales. Specifically, the local branch employs a 1×1 depthwise convolution to extract fine-grained features, whereas the large-scale branch captures long-range spatial dependencies via anisotropic large-kernel depthwise convolutions (63×1, 1×63, and 63×63). The global branch introduces a dual-domain channel attention module (DCAM) and a frequency-based spatial attention module (FSAM) to enhance the responses of the salient regions. Finally, a 1×1 convolution is applied for feature fusion, and a residual connection is adopted to stabilize training. This design enables global context modeling while maintaining computational efficiency.

The working principle of OmniKernel is illustrated in Figure 5.

 doi:10.20944/preprints202602.1454.v1

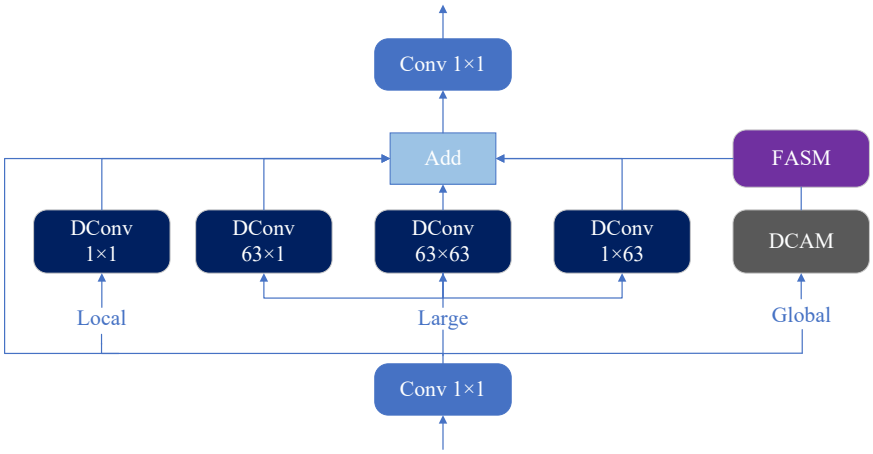

Figure 5. Schematic diagram of OmniKernel.

The convolution kernel set $K = \{3,5,7\}$ and the dilation rates $D = \{1,2,3\}$ jointly construct a dynamic receptive field, enabling the network to simultaneously capture local structural textures and long-range contextual dependencies. Unlike conventional unidirectional pyramid propagation, SOAPN introduces multilevel, weight-sharing cross-scale skip connections within the feature fusion pathway. This design is better suited to the rapidly changing viewpoints encountered in high-speed driving scenarios and effectively enhances the consistency of semantic and texture representations throughout the network.

3.3. TADAHead

In this paper, we further improve the detection head and propose a new TADAHead to solve several limitations of YOLOv8 in autonomous driving: low-texture feature adaptation, strong coupling between classification and regression, and poor spatial feature alignment. Conventional YOLO detection heads rely on shared feature representations to perform both classification and bounding box regression at the output, leading to semantic classification versus precise localization of feature representation. This problem becomes especially challenging in deep low-light scenarios (such as dense small-object distributions, heavy occlusion, or low-intensity conditions), where features in the same location must encode class semantics at the same time while maintaining sufficient sensitivity to boundary deformations, which leads to strong gradient competition and error coupling.

To solve these problems, TADAHead proposed multilevel dynamic task decomposition, learnable scale-aware weighting and dynamic offset-aligned convolution to model adaptive feature modeling, task decoupling and spatial recalibration of the task and to help the detection head obtain more discriminative task-specific representations at the prediction time.

Specifically, TADAHead first receives multiscale feature maps from the backbone and neck networks. Assuming that there are n_l input feature levels, the input can be denoted as $\mathbf{x} = \{x_1, x_2, \ldots, x_{n_l}\}$. Each scale feature is initially processed by a shared convolutional stack (share_conv), which consists of a two-stage convolutional structure. Group normalization [24] is employed instead of batch normalization [25] to ensure statistical stability under small-batch training conditions. This shared encoding extracts a unified base representation for both classification and regression tasks, reducing parameter redundancy while enhancing cross-scale feature consistency.

Next, we provide a task decomposition module that divides task-independent shared features into two task-specific branches, with global average pooling as additional guides. cls_decomp focuses on improving classification-related features, and reg_decamp improves localization-related representations. With this approach, the classification gradients do not affect the regression features of typical coupled detection heads. From a theoretical point of view, the suppression effect of high-

 doi:10.20944/preprints202602.1454.v1

variance positive GIoU samples by classification optimization can be avoided, and thus, the localization accuracy is improved.

To address insufficient spatial alignment caused by fixed convolutional sampling locations, DyDCNv2 dynamic deformable convolution is incorporated into the regression branch. This module combines the advantages of dynamic convolution [26] and deformable convolution [27] and introduces a spatial_conv_offset subnetwork to predict pixel-level offsets and modulation masks, enabling dynamic spatial feature resampling. The offset dimension is defined as $2 \times 3 \times 3$, corresponding to adaptive adjustment of the nine sampling points in deformable convolution. This design allows regression features to focus more effectively on object boundaries, corners, and fine-grained geometric regions. Compared with the fixed-kernel structure of the original YOLOv8 head, the proposed approach significantly improves robustness under occlusion, scale variation, and shape deformation.

In addition, a CLS-Prob capability modulation mechanism is proposed to further enhance classification performance. This module employs cls_prob_conv1 and cls_prob_conv2 to predict classification confidence priors, which are activated via a sigmoid[28] function and then applied to the classification feature maps via elementwise multiplication. This attention-guided semantic modulation effectively suppresses low-confidence noisy activations and enhances class discrimination ability.

At the output stage, TADAHead generates distributional predictions of size $4 \times \text{reg\_max}$ using the cv2 layer, which are decoded into continuous bounding box coordinates via distribution focal loss (DFL). The decoded predictions are then combined with dynamically generated anchors and strides to recover absolute-scale bounding boxes. During inference, predictions from all feature levels are aggregated into a unified output tensor of the form

$$Y = [x, y, w, h, p_1, p_2, \dots, p_{nc}],$$

where the first four elements represent bounding box coordinates and the remaining ncelements correspond to class probabilities. TADAHead supports both training and deployment modes and is compatible with lightweight inference frameworks such as TFLite and EdgeTPU, enabling direct application in embedded and real-time autonomous driving systems.

The overall architecture of TADAHead is illustrated in Figure 6.

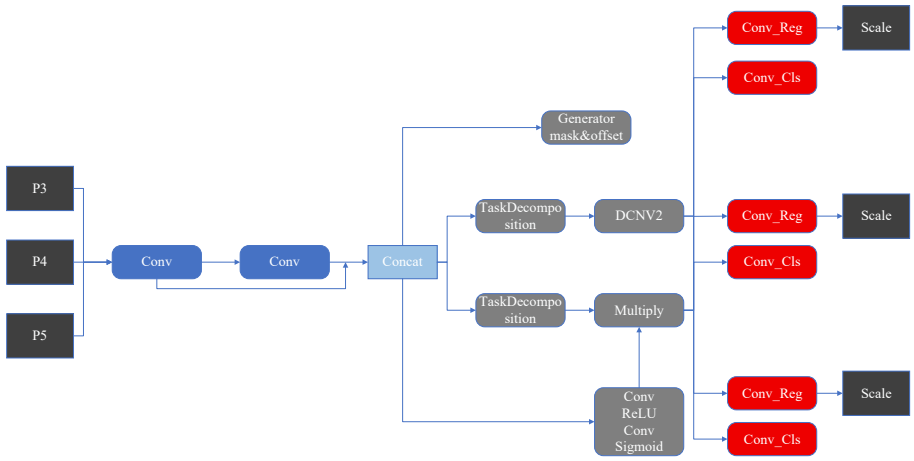

Figure 6. Structural diagram of TADAHead.

4. Experiments And Analysis

4.1. Dataset

The dataset used in this study is derived from the publicly available KITTI benchmark dataset [29]. KITTI was jointly established by the Karlsruhe Institute of Technology (KIT) and the Toyota Technological Institute at Chicago (TTIC) and has become one of the most influential large-scale

 doi:10.20944/preprints202602.1454.v1

evaluation benchmarks for computer vision algorithms in autonomous driving scenarios. The dataset contains real-world images collected from urban, rural, and highway environments. Each image may include up to 15 vehicles and 30 pedestrians, accompanied by varying degrees of occlusion and truncation. The annotated categories cover eight classes, namely, Car, Van, Truck, Pedestrian, Person Sitting, Cyclist, Tram, and Misc. Owing to its comprehensive coverage of common autonomous driving object categories, the KITTI dataset is well suited for evaluating the detection accuracy of the proposed model.

In this work, a total of 7,481 annotated images from the KITTI dataset are selected, with representative examples shown in Figure 7. Three frequently occurring object categories—Car, Pedestrian, and Cyclist—are used for experimental evaluation.To increase data diversity and improve the generalization ability of the model, various geometric transformations are applied to the images for data augmentation. Finally, the dataset is split into training, validation, and test sets at a ratio of 8:1:1.

Figure 7. Sample images from the dataset.

4.2. Experimental Environment And Training Settings

All the experiments are run on Linux. The computer consists of an AMD Ryzen 7 7745HX CPU, 16 GB of RAM, and an NVIDIA GeForce RTX 4090 GPU with 24 GB of video memory. The operating system used was Python 3.8, PyTorch 2.2.2, and CUDA 12.1. The experimental environment is summarized in Table 1.

Table 1. Experimental environment.

Experimental Environment	Value
Processor	R7-7745HX
Operating System	Linux
Memory	16GB
GPU	RTX 4090
GPU Memory	24GB
Programming Language	Python3.8
Deep Learning Framework	PyTorch2.2.2
Deep Learning Toolkit	CUDA12.1

This study uses YOLOv8s as the baseline model. The input image resolution is 640 × 640. The SGD optimizer is trained at the 1st learning rate of 0.01, and the cosine annealing strategy is applied at the 1st training rate of 0, momentum of 0.9 and weight decay coefficient of 0.005. The batch size is 32, and the model is trained for 250 epochs. The training hyperparameter settings are shown in Table 2.

Table 2. Training parameter settings.

Parameter	Value
Input Image Size	640×640

 doi:10.20944/preprints202602.1454.v1

Learning Rate	0.01
Weight Decay	0.005
Momentum	0.9
Optimizer	SGD
Batch Size	32
Training Epochs	250

4.3. Evaluation Metrics

To measure the performance of our model, several popular metrics, such as precision (P), recall (R), mean average precision (mAP), and frames per second (FPS), are also used. These metrics are defined and computed as follows.

The precision (P) is the ratio of correctly predicted positive samples to all predicted positive samples, which can be interpreted as

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

where TP denotes true positives and where FP denotes false positives.

Recall that (R) is the ratio of correctly predicted positive samples to all real positive samples:

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

where FN denotes false negatives.

The F1 score is the harmonic mean of precision and recall reflecting the performance of the model, and the following is the F1 score:

$$F_1 = \frac{1}{\frac{1}{R} + \frac{1}{P}} = \frac{2 \times P \times R}{P + R}$$

mAP (mean average precision) is the performance of all classes. AP provides detection accuracy for one class and mAP for all categories. The formulas are as follows:

$$AP = \sum_{i=1}^{n} (R_i - R_{i-1}) \cdot P_{i-1}$$

$$mAP = \frac{1}{C} \sum_{s=1}^{C} AP_s$$

where n is the number of recall levels, C is the total number of objects, and APs is the average precision of the s-th class.

In addition, FPS can be used to evaluate the real-time performance of the model with the inference speed in real-world applications.

4.4. Comparative Experiments

To objectively verify the reliability and effectiveness of the proposed SST-YOLO model in autonomous driving object detection tasks, several representative object detection algorithms, including SSD, Faster R-CNN, YOLOv3, YOLOv5, YOLOv7, YOLOv8, and YOLOv10, were selected for comparison. All compared models were trained and evaluated under the same experimental settings to ensure fairness. The quantitative comparison results are reported in Table 3.

Table 3. Comparison Results of Different Object Detection Models.

Model	P/%	R/%	mAP@0.5/%	mAP@0.5-0.95/%
SSD	86.4	79.4	83.3	77.9
Faster RCNN	84.5	77.7	80.9	67.5
YOLOv3	89.3	83.3	86.9	64.7
YOLOv5	90.3	84.7	88.6	65.5
YOLOv7	90.5	84.1	87.4	64.5
YOLOv8	90.1	82.9	88.1	65.3

 doi:10.20944/preprints202602.1454.v1

YOLOv10	91.1	85.2	88.9	65.7
SST-YOLO	93.4	84.4	91.7	69.2

As shown in Table 3, the proposed SST-YOLO achieves an mAP@0.5 of 91.7%, which is higher than those of SSD, Faster R-CNN, YOLOv3, YOLOv5, YOLOv7, YOLOv8, and YOLOv10 by 8.5%, 10.8%, 4.8%, 3.1%, 4.3%, 3.6% and 2.8%, respectively. SST-YOLO has the highest precision (93.4%) and competitive recall (84.4%), indicating higher sensitivity to small-scale and high-level targets. These results validate the performance of the proposed improvements and demonstrate the strong detection power of SST-YOLO over SST-YOLO in autonomous driving scenarios.

4.5. Ablation Experiments

To check the performance of each improvement module, this study conduct ablation experiments and present the results in Table 4. YOLOv8s+SCC improves the backbone of YOLOv8s by replacing the bottleneck module in C2f with the proposed SCC module. Compared with the baseline YOLOv8s, the results yield mAP@0.5, mAP@0.5–0.95 and FPS results. The SCC provides rich edge information and spatial information of the learned features to explain more image content and detection accuracy.

On this basis, YOLOv8s+SCC+SOAPN further enhances the neck of YOLOv8s by replacing the original PAFPN with the proposed SOAPN structure. Compared with the previous configuration, consistent gains are observed in mAP@0.5, mAP@0.5–0.95, and FPS, demonstrating that SOAPN is capable of effectively learning feature representations from global semantics to local details, which significantly benefits small object detection performance.

Additionally, YOLOv8s+SCC+SOAPN+TADAHead replaces the detection head with the proposed TADAHead. In addition to the aforementioned model, all the evaluation metrics show that TADAHead significantly reduces the multitask modeling conflict between classification and regression. In a related direction, the proposed dynamic convolution facilitates spatial feature adaptability, which increases detection accuracy and robustness in challenging autonomous driving scenarios.

Table 4. Ablation experiment results.

Model	P/%	R/%	mAP@0.5/%	mAP@0.5–0.95/%
YOLOv8s	88.9	82.7	89.2	65.1
YOLOv8s+SCC	92.5	80.8	90.2	65.5
YOLOv8s+SCC+SOAPN	93.2	82.7	91.6	68.6
YOLOv8s+SCC+SOAPN+TADAHead	93.4	84.4	91.7	69.2

4.6. Model Generalization Experiments

The SODA10M dataset [30], which was jointly released by Huawei Noah's Ark Lab and Sun Yat-sen University, is a new-generation semi/self-supervised 2D benchmark dataset for autonomous driving. The images are collected from 32 different cities, covering most regions of China, and include a wide variety of driving scenarios, such as urban roads, highways, suburban roads, and industrial parks. In addition, the dataset spans diverse weather conditions (sunny, cloudy, rainy, and snowy) and time periods (daytime, nighttime, and dawn/dusk), providing strong diversity and complexity. Owing to these characteristics, SODA10M is well suited for evaluating the generalization capability of object detection models.

A total of 10,000 annotated images from the SODA10M dataset are selected to fit the training, validation and test sets at an 8:1:1 ratio. The trained models are compared on the test set for generalization. To investigate the external validity of the proposed SST-YOLO, comparisons with several popular object detectors are conducted, and the quantitative results are shown in Table 5.

As shown in Table 5, SST-YOLO achieves the best accuracies, with mAP@0.5 and mAP@0.5–0.95, with accuracies of 81.4%, 87.8% and 60.7%, respectively. While SST-YOLO provides a lower recall

 doi:10.20944/preprints202602.1454.v1

than YOLOv8 does, its precision and mAP are more comparable to those of YOLOv8. Compared with YOLOv7, SST-YOLO results in lower recall and higher scores for other evaluation metrics. Compared with YOLOv5, SST-YOLO yields better performance in terms of accuracy, recall, mAP@0.5, and mAP@0.5–0.95. These results show that SST-YOLO offers competitive and stable performance across all driving environments and proves its generalizability and external performance.

Table 5. Model generalization experiment results

Model	P/%	R/%	mAP@0.5/%	mAP@0.5–0.95/%
YOLOv5	79.2	71.3	83.1	55.1
YOLOv7	81.1	73.7	82.1	54.6
YOLOv8	80.9	73.8	82.3	56.4
SST-YOLO	81.4	73.3	87.8	60.7

4.7. Visualization Analysis

To further verify the reliability of SST-YOLO in detecting small objects in autonomous driving scenarios, qualitative visualization analysis is conducted, as illustrated in Figure 8. In the figure, the first column shows the detection results of the YOLOv8 baseline model, whereas the second column presents the results obtained by the proposed SST-YOLO.

As observed in the first row, SST-YOLO achieves a confidence score that is approximately 0.2 higher than that of the baseline model for the vehicle located on the far right of the image. In the second row, SST-YOLO successfully detects a small-sized vehicle in the left region of the image that is missed by the baseline model. In the third row, the confidence score for the rightmost vehicle detected by SST-YOLO exceeds that of YOLOv8 by approximately 0.6.

These visualization results show that this SST-YOLO not only achieves more accurate detection than the baseline but also has higher detection accuracy for small and easily missed targets (a large part of autonomous driving).

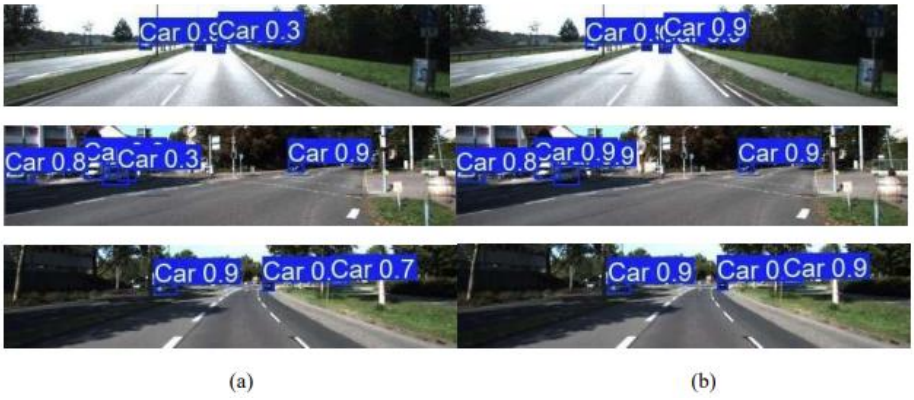

Figure 8. Visualization analysis of the experimental results.(a) Images detected by YOLOv8 (b) Images detected by SST-YOLO.

5. Conclusion

This paper propose an object detection model for autonomous driving called SST-YOLO. On the basis of YOLOv8, implement SCC in the backbone, replace the neck with SOAPN, and deploy the TADAHead detector head

The SCC module concatenates the features acquired with SobelConv and standard convolution, allowing the learned representations to capture all edge information and spatial semantic features to accurately describe the image content. SOAPN generalizes bidirectional feature representations by employing heterogeneous-scale convolutions, multidimensional receptive field fusion and lightweight cross-layer aggregation, thereby increasing global semantic information and preserving local detail. As a result, the detection network is more sensitive to small targets, has better long-distance discrimination ability and is more robust to occlusion in autonomous driving. TADAHead

 doi:10.20944/preprints202602.1454.v1

improves detection performance by using multilevel dynamic decomposition, learnable scale weights and dynamic offset aligned convolutions with feature adaptivity, task decoupling and spatial recalibration, which effectively improves the discriminative performance of the detection head.

Experiments show that SST-YOLO achieves an mAP@0.5-0.95 of 68.2% on the autonomous driving object detection task, outperforming the baseline model and meeting the requirements of autonomous driving perception.

The work in the future will also improve the robustness of autonomous driving object detection algorithms under challenging conditions (microsensor noise, strong light, and backlighting) and model lightweighting and in-vehicle deployment optimization, which will lead to more widespread application of autonomous vehicle object detection.

Author Contributions: Ningbo Zhang&Qinsheng Du are responsible for the design and execution of the experiments, as well as the writing of the manuscript.Wenqing Bi,Ruidi Zhu,Yuhan Liu are responsible for preparing the figures and tables.Chao Shen&Shiyan Zhang are responsible for organizing the experimental data.Jian Zhao is responsible for providing experimental equipment and technical support. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data availability Statement: The datasets used and analyzed during the current study are publicly available and can be accessed from http://www.semantic-kitti.org/dataset.html#download.

Acknowledgments: We would like to express our deepest gratitude to all those who have contributed to the completion of this research and the writing of this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yurtsever, E., Lambert, J., Carballo, A., & Takeda, K. (2020). A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*.**8,** 58443-58469.
2. Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*.**2018**(1), 7068349.
3. Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). Object detection in 20 years: A survey. *Proceedings of the IEEE*. **111**(3), 257-276.
4. Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., & Bennamoun, M. (2020). Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, **43**(12), 4338-4364.
5. Liu, Y., Sun, P., Wergeles, N., & Shang, Y. (2021). A survey and performance evaluation of deep learning methods for small object detection. *Expert Systems with Applications*.**172,** 114602.
6. Yang, X., Wu, W., Liu, K., Kim, P. W., Sangaiah, A. K., & Jeon, G. (2018). Long-distance object recognition with image super resolution: A comparative study. *IEEE Access*.**6,** 13429-13438.
7. LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature.**2015**, 521(7553): 436-444.
8. Alzubaidi. et al. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*.**8**(1), 53.
9. Khan, F. A., Gumaei, A., Derhab, A., & Hussain, A. (2019). A novel two-stage deep learning model for efficient network intrusion detection. *Ieee Access*.**7,** 30373-30385.
10. Wang, T., Yang, F., & Tsui, K. L. (2020). Real-time detection of railway track component via one-stage deep learning networks. *Sensors*.**20**(15), 4325.
11. Chen, L. et al. (2024). End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.**46**(12), 10164-10183.
12. Kanopoulos, N., Vasanthavada, N., & Baker, R. L. (1988). Design of an image edge detection filter using the Sobel operator. *IEEE Journal of solid-state circuits*.**23**(2), 358-367.
13. Bappy, J. H., & Roy-Chowdhury, A. K. (2016, September). CNN based region proposals for efficient object detection. In *2016 IEEE International Conference on Image Processing (ICIP)* (pp. 3658-3662). IEEE.

 doi:10.20944/preprints202602.1454.v1

14. Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).
15. Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*.**39**(6), 1137-1149.
16. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, September). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Cham: Springer International Publishing.
17. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).
18. Uijlings. et al. (2013). Selective search for object recognition. *International journal of computer vision*.**104**(2), 154-171.
19. Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10781-10790).
20. Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8759-8768).
21. Sunkara, R., & Luo, T. (2022, September). No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 443-459). Cham: Springer Nature Switzerland.
22. Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 390-391).
23. Cui, Y., Ren, W., & Knoll, A. (2024, March). Omni-kernel network for image restoration. In Proceedings of the AAAI conference on artificial intelligence (Vol. 38, No. 2, pp. 1426-1434).
24. Wu, Y., & He, K. (2018). Group normalization. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 3-19).
25. Bjorck, N., Gomes, C. P., Selman, B., & Weinberger, K. Q. (2018). Understanding batch normalization. Advances in neural information processing systems, 31.
26. Chen. et al. (2020). Dynamic convolution: Attention over convolution kernels. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11030-11039).
27. Zhu, X., Hu, H., Lin, S., & Dai, J. (2019). Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9308-9316).
28. Kyurkchiev, N., & Markov, S. (2015). Sigmoid functions: some approximation and modelling aspects. *LAP LAMBERT Academic Publishing, Saarbrucken*.**4,** 34.
29. Behley. et al. (2019). Semantickitti: A dataset for semantic scene understanding of lidar sequences. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 9297-9307).
30. Han, J. et al. (2021). SODA10M: A large-scale 2D self/semi-supervised object detection dataset for autonomous driving. arXiv preprint arXiv:2106.11118.