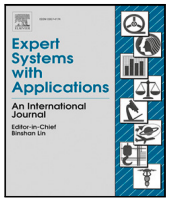
Expert Systems With Applications 256 (2024) 124942

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



# SD-YOLO-AWDNet: A hybrid approach for smart object detection in challenging weather for self-driving cars

Rashmi, Rashmi Chaudhry *

*Netaji Subhas University of Technology, New Delhi, India*

ARTICLE INFO

*Keywords:*
Object detection
Self driving cars
Deep neural network
YOLO

ABSTRACT

Several deep learning algorithms are currently focused on object detection in adverse weather scenarios for autonomous driving systems. However, these algorithms face challenges in real-time scenarios, leading to a reduction in detection accuracy. To tackle these issues, this paper introduces a lightweight object detection model named Self Driving Cars You Only Look Once Adverse Weather Detection Network (SD-YOLO-AWDNet), derived from enhancements to the YOLOv5 algorithm. The model incorporates four progressive improvement levels within the YOLOv5 framework. This includes integrating C3Ghost and GhostConv modules in the backbone to enhance detection speed by reducing computational overhead during feature extraction. To address potential accuracy issues arising from these modules, Depthwise-Separable Dilated Convolutions (DSDC) are introduced, striking a balance between accuracy and parameter reduction. The model further incorporates a Coordinate Attention (CA) module in the GhostBottleneck to enhance feature extraction and eliminate unnecessary features, improving precision in object detection. Additionally, a novel "Focal Distribution Loss" replaces CIoU Loss, accelerating bounding box regression and loss reduction. Test dataset experiments demonstrate that SD-YOLO-AWDNet outperforms YOLOv5 with a 54% decrease in FLOPs, a 52.53% decrease in model parameters, a 2.24% increase in mAP, and a threefold improvement in detection speed.

## 1. Introduction

As per a technical report by the National Highway Traffic Safety Administration (NHTSA), nearly 94% of accidents on the road are attributed to human mistakes (Simhambhatla, Okiah, Kuchkula, & Slater, 2019). In addressing this issue, Self-Driving Systems (SDS) have been created to prevent accidents, alleviate driving-related stress, lower emissions, and offer accessible transportation for individuals with mobility impairments (Geiger, Lenz, & Urtasun, 2012).

These systems are sophisticated and intelligent, handling a variety of responsibilities across different phases from perception to planning (Geiger et al., 2012; Wang, Wen, Wang, Huang, & Pei, 2019). Fig. 1 depicts the comprehensive classification of stages in self-driving cars.

The perception phase involves identifying objects and motion in the environment, encompassing static object detection like roads, lane markings, construction signs, and obstructions, as well as dynamic object detection including vehicles and pedestrians. Furthermore, ego localization entails estimating our position, velocity, acceleration, orientation, and angular motion at any given time.

The planning phase includes decisions related to the driver's tasks which can be classified as reactive rule-based planning and predictive planning. Reactive planning includes rules that take into account the current state of the ego vehicle and other objects to make decisions. Predictive planning is about other vehicles and how they are moving (Wang et al., 2019).

This paper addresses a specific challenge within the domain of the Perception Phase of self-driving or autonomous vehicles, concentrating on the objective of detecting objects in challenging weather conditions. The Perception Phase involves various challenges, such as robust detection and segmentation, sensor uncertainty leading to potentially corrupted GPS measurements, noise in LiDAR/RADAR readings, occlusion, reflection, illumination, lens flare, and precipitation (Johari & Swami, 2020). The impact of weather conditions on detection accuracy is contingent upon the quality of input data from sensors. Input data plays a pivotal role in ensuring higher security levels for Self-Driving Systems and Self-driving vehicles (Hnewa & Radha, 2020). Consequently, contemporary self-driving vehicles leverage cameras, sensors, and deep learning approaches for object classification and

* Corresponding author.
*E-mail addresses:* rashmi.phd21@nsut.ac.in (Rashmi), rashmi@nsut.ac.in (R. Chaudhry).

https://doi.org/10.1016/j.eswa.2024.124942
Received 17 January 2024; Received in revised form 13 July 2024; Accepted 28 July 2024
Available online 31 July 2024

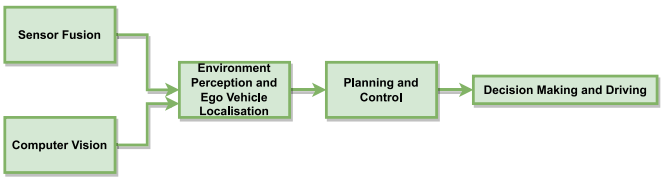



**Fig. 1.** General taxonomy of perception and planning in Autonomous driving (Johari & Swami, 2020).

detection (Hnewa & Radha, 2020). Fig. 2 illustrates the object detection scene of BDD100k datasets (Yu et al., 2020) for self-driving cars.

Fig. 3 shows various adverse weather conditions and highlights the challenges. Weather conditions like heavy rain, snow or fog can reduce visibility, which makes it hard to detect objects' sizes and positions. This may lead to inaccuracy of the object detection system. Such weather conditions can also cause partial or complete object occlusion. Traditional object-detecting algorithms may struggle with these visibility and occlusion issues.

Deep learning-based algorithms have been dominating the landscape, showcasing advancements in getting better at limitations with obscured objects and improving overall understanding of the scene. With the continuous evolution of Convolutions Neural Network (CNN) models, it has gotten more popular, as evidenced by recent developments (Hnewa & Radha, 2020).

Object detection algorithms are generally classified into 2 categories one-stage or two-stage detectors. Over the past few decades, researchers have made significant improvements in both categories.

One-stage detectors are very well suited for real-time applications because of their speed and streamlined architecture. These models can directly predict object locations and classes without the need for an explicit region proposal step (Tian, Shen, Chen & He, 2019). Alternatively, two-stage detectors prioritize precision, employing a sequential two-phase approach (Redmon, Divvala, Girshick, & Farhadi, 2016; Wang, Bochkovskiy & Liao, 2023). This involves initially generating region proposals, followed by meticulous refinement of object localization and classification (Li et al., 2017). Deciding between these two detection approaches relies on the specific needs of the application in question, necessitating a thoughtful evaluation to find an equilibrium between computational speed and detection accuracy.

Adverse weather conditions, including fog, rain, haze, snow, mist, and thunderstorms, can notably reduce the quality of visual data collected by sensors, cameras, and LiDAR systems. This often leads to decreased visibility and image fidelity, consequently degrading the performance of object detection. In particular, the vehicle may encounter difficulties in identifying crucial objects within its surroundings. The different scenarios of adverse weather conditions are shown in Fig. 3. These can be classified into two categories namely dynamic (rain and snow) or steady (mist, fog, and haze) weather conditions. Dynamic weather conditions are more complex than visual effects. Various studies are focusing on various types of dynamic weather conditions independently. In addressing the challenges posed by both dynamic and steady weather conditions, the various studies are focusing on distinct types independently.

Recognizing the complexities of adverse weather conditions, this paper proposes a novel model, SD-YOLO-AWDNet, which integrates ghost convolutions and an enhanced loss function into YOLOv5. This innovative approach aims to significantly improve detection performance, providing a robust solution for adverse weather scenarios. Key contributions of the paper are given as follows:

1. Novel SD-YOLO-AWDNet model is proposed based on YOLOv5 for adverse weather scenarios in self-driving cars. This model incorporates a novel loss function known as the "Focal-Distribution Loss", which is designed to expedite the convergence rate of the model.
2. Moreover, Within the YOLOv5 backbone, traditional convolutional layers were replaced by lightweight C3Ghost and GhostConv modules, achieving model compression via cost-effective linear operations while maintaining detection speed and accuracy.
3. New Depthwise separable Dilated Convolutions (DSDC) with residual structures are introduced to improve feature extraction capabilities while maintaining computational costs.
4. The Coordinated Attention (CA) module was included to improve the extraction of relevant features while suppressing irrelevant ones. This augmentation aims to boost the accuracy of the algorithm's detection capabilities.
5. The performance is compared with various Tensorflow Object Detection (TFOD) models, including two-stage detectors (Faster RCNN, ResNet 101 Girshick, 2015), and one-stage detectors (SSD ResNet v1 FPN Liu et al., 2016, CenterNet (Duan et al., 2019), YOLOv5 (Jocher et al., 2022)), YOLOv6 (Li, Li, et al., 2022), YOLOv7 (Wang, Bochkovskiy & Liao, 2023), YOLOv7-tiny (Cheng et al., 2023), and YOLOv8 (Vats & Anastasiu, 2023).

These models are trained on the BDD100k dataset (Yu et al., 2020), intentionally curated to encompass a wide array of corner cases, thereby comprehensively addressing adverse scenarios such as fog, snow, rain, sunshine, and low-light conditions. Additionally, selected images of above mentioned adverse scenarios from the BDD100k dataset (Yu et al., 2020) are examined independently. While primary analysis focuses on BDD100k dataset due to its extensive nature, additional validation is conducted on the CADC (Pitropov et al., 2021) and KITTI (Geiger, Lenz, Stiller, & Urtasun, 2013) datasets to ensure performance consistency across varied datasets.

## 2. Related work

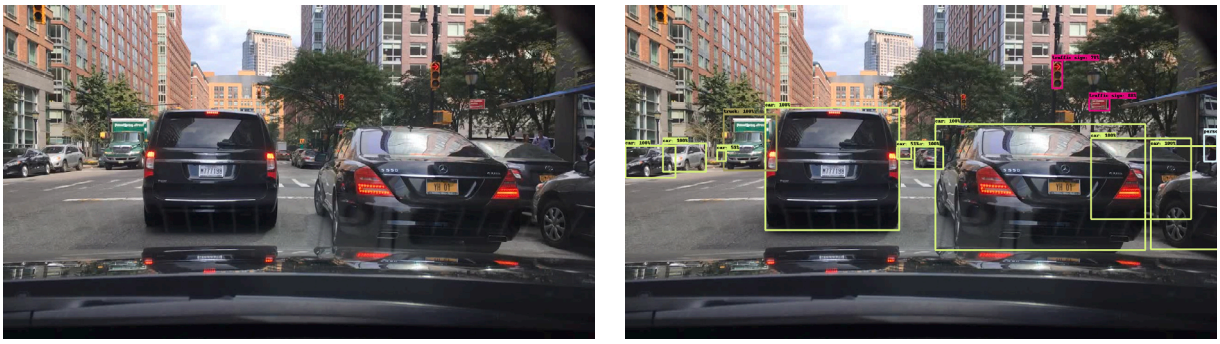
Over the past few years, there is been a noticeable increase in attention directed towards object detection tasks, especially within the domain of autonomous vehicles, which plays an increasingly prominent role in advancing autonomous driving technology (Paz, Zhang, Li, Xiang, & Christensen, 2020). Consequently, object detection techniques under severe weather conditions have arisen as a pivotal and substantial area of exploration in research (Cai et al., 2021).

Deep learning uses deep neural networks for object detection. However, the input of these networks may be affected by different adverse weather conditions including too sunny, snow, heavy rain, and haze, etc which leads to wrong predictions or classification of objects on the road (Ouyang et al., 2015).

However, various studies have worked on this problem to provide proper assessment. Some study focuses on the usage of both camera and radar information fusion sensing methods instead of using a single radar sensor in adverse weather conditions (Liu et al., 2021). Some are working on CARLA simulator for depth estimation along with 2d object detection (Tabata, Zimmer, dos Santos Coelho, & Mariani, 2023).

Some researchers concentrate their efforts solely on data sets affected by snowfall, employing 2D object detection methods and addressing issues related to missing data points caused by snow (Michaelis et al., 2019). Others adopt an Image adaptation approach, which adaptively enhances images to improve detection capabilities, as demonstrated in Liu et al. (2022). Most studies in this domain tend to isolate specific adverse weather parameters such as snow or fog and often employ specialized datasets like VOC foggy (Sakaridis, Dai, & Van Gool, 2018) and RTTS (Wang et al., 2022). While some concentrate on deraining images for rainy weather conditions (Jiang, Zhu, Zhao, & Deng, 2023).

One conventional approach involves pre-processing the images using classical Dehazing or image enhancement techniques, as discussed in Liu, Ma, Shi, and Chen (2019). These methods were initially developed to alleviate foggy conditions and enhance overall image quality.



((a)) Original image ((b)) Predicted image

**Fig. 2.** Object Detection on a scene of traffic.

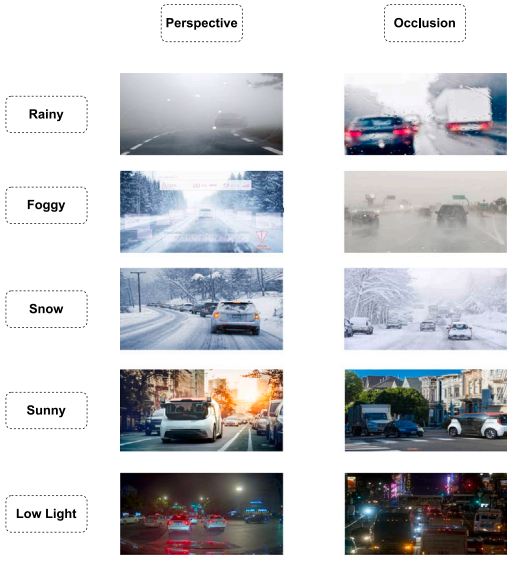



**Fig. 3.** General Scenarios of Adverse Weather conditions in Autonomous Driving.

Nevertheless, it is important to note that improving image quality may not always guarantee a corresponding enhancement in detection performance.

Alternatively, some researchers are exploring innovative approaches, such as Sigma: Semantic-complete graph matching, to address challenges in object detection domain adaptation, as outlined in Li, Liu, and Yuan (2022).

Researchers also covered 3D object detection (Rezaei, Azarmi, & Mir, 2023) but are focusing on traffic monitoring rather than adverse weather scenarios. Some Case studies also work with 6d pose estimation but are yet to be implemented in adverse weather scenarios (Hoque, Xu, Maiti, Wei, & Arafat, 2023).

While the majority of the research has concentrated on general object detection (Wang, Xu, et al., 2023), there has been relatively limited exploration into object detection under adverse weather conditions. One prevalent and uncomplicated strategy involves domain-adaptive object detection designed specifically for autonomous vehicles in misty conditions environments, with a significant focus on improving detection capabilities in foggy settings, as outlined in Li et al. (2023).

Numerous prior methodologies (Zhang et al., 2021) have utilized combined image amplification and detection techniques to alleviate the influence of adverse weather information, specifically.

In another study, an unsupervised domain adversarial object detection framework based on prior knowledge for object detection in rainy and hazy conditions was introduced by Sindagi, Oza, Yasarla, and Patel (2020).

Another set of methods, as demonstrated in Hnewa and Radha (2021), tackle this issue through domain adaptation, Assuming a domain shift between images taken under regular and adverse weather conditions. They developed a multi-scale domain adaptive YOLO model that facilitates the adaptation of domains at various layers during the process of extracting features.

Certain approaches in the field of computer vision for autonomous driving concentrate on specialized object detection tasks, exemplified by Tian, Gelernter, Wang, Li and Yu (2019), such as precise identification of traffic lights and signs. Some approaches emphasize object feedback and feature retention to enhance small object detection efficacy (Tian, Han, & Wang, 2024). In another study, the focus is on detecting small objects, particularly addressing multiscale defects, within complex background interference scenarios, aiming to improve object detection precision in challenging environments (Zhang, Zhang, Huang, Han, & Zhao, 2024).

Domains like traffic monitoring and smart parking pose challenges for real-time vehicle detection and classification, including the identification of small-sized objects and variations in UAV altitude and angles. Hamzenejadi and Mohseni (2023) specifically addresses vehicle detection, emphasizing small-sized objects in these scenarios. Another study explores strategies to reduce computational burden while maintaining accuracy, though it does not cater specifically to self-driving car scenarios (Wang, Li, Liu, & Meng, 2024). Notably, none of these studies have focused on adverse weather conditions affecting self-driving cars.

While many studies tend to narrow their focus on specific weather parameters like snow or fog, often utilizing datasets such as VOC foggy (Sakaridis et al., 2018) and RTTS (Sindagi et al., 2020), this paper adopts a broader perspective.

Lightweight networks primarily adopt two key strategies. The first involves pruning the target model network structure, for instance, YOLOv3-tiny (Adarsh, Rathi, & Kumar, 2020) and YOLOv4-tiny (Wang, Bochkovskiy, & Liao, 2021) tailored for YOLOv3 and YOLOv4, respectively. The second method involves adapting network architecture by adjusting depth or width scaling factors, as seen in EfficientDet by Tan, Pang, and Le (2020) and Ge, Liu, Wang, Li, and Sun (2021), offering variants like YOLOX-nano and YOLOX-tiny.

Mahaur, Mishra, and Kumar (2023) introduced a model improving the detection performance of small-scale objects through network pruning, reducing computational costs. Another study focused on small object detection by adding a detection layer for finer feature maps in the neck network pyramid and integrating the C3CrossConv module into the backbone network (Gao, Ji, Yu, & Yuan, 2024). A lightweight object detection model, YOLOv8-Lite, based on the YOLOv8 framework and enhanced with the FastDet structure, TFPN pyramid, and CBAM attention mechanism, was proposed (Yang & Fan, 2024). Enhancements in YOLOv5 include the convolutional block attention mechanism for channel and spatial feature attention, the slimming pruning algorithm for improved efficiency, and knowledge distillation for accuracy compensation during fine-tuning (Li, Zhuang, Bao, Chen, & Yang, 2024).

For instance, Point-GNN is designed to predict the category and shape of the object each vertex belongs to, with an auto-registration mechanism to reduce translation variance and a box merging and scoring operation for accurate detections (Shi & Rajkumar, 2020). A GNN-based feature extraction of point cloud maps was constructed to increase the receptive fields of learning map features, based on PVRCNN (Liao, Wang, & Lin, 2023). However, these studies focus on normal scenarios, not adverse weather conditions. While these models are integrated into autonomous driving systems, their improved performance on less powerful hardware often compromises detection accuracy (Chen, Ding, Zou, Chen, & Li, 2020). Conversely, larger models offer better accuracy but require extensive computation. Recent studies employ dilated and depth-wise separable convolutions to mitigate accuracy loss in lightweight models (Sun, Zhang, & He, 2020). Additionally, coordinate attention mechanisms have been investigated for detecting small objects (Xuan et al., 2022).

However, challenges remain in adapting these techniques for practical development, such as in autonomous driving systems. This paper addresses such challenges by proposing a lighter model structure with fewer parameters, ensuring better precision efficiency.

Here, the research centers around various TensorFlow-based object detection models, comparing their performance across a cross-weather hybrid dataset of BDD100k (Yu et al., 2020) to handle diverse weather scenarios. It covers a range of adverse weather conditions, including cloudiness, rainfall, snowfall, bright sunshine, sandy environments, and sunrise conditions. Validation testing is done on popular datasets of CADC (Pitropov et al., 2021) and KITTI (Geiger et al., 2013) as well.

## 3. Model introduction and improvement

In this section, multiple enhancements are suggested for the traditional YOLOv5 model, i.e, Light-weighted model structure substituting the Conv + BatchNorm + SiLU (CBS) and Cross Stage partial (CSP) modules with GhostConv and C3Ghost modules, incorporating the Coordinate Attention (CA) mechanism (Hou, Zhou, & Feng, 2021), adding Depthwise Separable Dilated Convolutions (DSDCs), and swapping out the Complete Intersection over Union (CIoU) loss with "Focal-Distribution Loss" in the loss function. The resulting model, depicted in Fig. 4, is denoted as SD-YOLO-AWDNet. The subsequent sections elaborate on each component of the network and their contribution towards the performance achieved.

### 3.1. YOLOv5 model introduction

The YOLOv5 variants, labelled n, s, m, l, and x, exhibit distinct network depth and width attributes, functioning as a one-stage target detection algorithm. The YOLOv5 model comprises four essential components: input, backbone network, neck, and the detection head. YOLOv5n boasts the least number of parameters and the quickest detection speed among these models. Nevertheless, its detection accuracy is comparatively lower when performance features are considered. While, YOLOv5x employs the largest parameter count and scans at a slower pace, despite achieving the highest detection accuracy. While the architecture remains consistent across all these models, variations in network depths and feature map widths enable scalable adjustments to the network model. YOLOv5s are more suited for object detection in autonomous vehicles based on the distinctive attributes of these versions. This research paper proposes an object detection technique for self-driving cars that balances network speed with detection precision by improving YOLOv5s.

The CBS module, Concentrated-Comprehensive Convolut (C3) module, and Spatial Pyramid Pooling F (SPPF) module make up the backbone network for YOLOv5s. CBS module combines three tasks: batch normalization, activation, and convolution (Conv2d), while the auto pad technique is utilized to create the full effect. Leaky ReLU is replaced with Swish (or SiLU) (Ramachandran, Zoph, & Le, 2017) as the activation function. By adapting CSPNet's (Han et al., 2020) concepts to the DarkNet53 backbone network, YOLOv5 extends and incorporates these principles. The YOLOv5s Neck incorporates concepts from the Path Aggregation Network (PANet) and Feature Pyramid Network (FPN) (Liu, Qi, Qin, Shi, & Jia, 2018). The positional information from shallow layers is incapable of influencing features at deeper levels. As a solution, PANet introduces a bottom-up channel a top FPN to transmit semantic information from deep to shallow layers. In this manner, the deep layer can also receive position data from the bottom layer, improving multi-scale localization capacity. These two structures, together, improve the neck network's ability to fuse features.

In YOLOv5s, the Head multiplies the number of channels by a $1 \times 1$ convolution for each of the various scales of feature maps obtained in the Neck. Additionally, the number of anchors on each detection layer is determined by (number of categories + 5), where 5 represents the width, height, and confidence of the centroid of the prediction frame, with confidence indicating the frame's confidence. Each feature map incorporates three pre-defined anchors with distinct aspect ratios for prediction. YOLOv5 partitions the grid based on the pixel sizes of the three different feature map scales. As a result, the channel dimension encapsulates all location and classification data associated with the anchor from the preceding frame within the feature map's channel dimension.

In YOLOv5's backbone, the CBS and CSP modules efficiently extract image features but contribute to a high number of model parameters, resulting in increased computational demands. To address this, the enhanced backbone incorporates GhostConv and C3Ghost modules, effectively reducing model parameters and computational complexity.

### 3.2. Model structure improvement

#### 3.2.1. Lightweighted model structure

To minimize model parameters, reduce model costs, and enhance overall accuracy, the design of SD-YOLO-AWDNet is introduced. The subsequent section outlines four components of architectural enhancements in SD-YOLO-AWDNet. SD-YOLO-AWDNet demonstrates promising strategies for reducing model size while potentially maintaining or improving accuracy.

Fig. 4 visually represents the improved SD-YOLO-AWDNet architecture. The diagram showcases the integration of GhostConv, C3Ghost, and CA modules, emphasizing the structural changes made to the backbone. The visual representation aids in understanding the architectural modifications made for a more efficient and accurate YOLOv5s model.

In YOLOv5's backbone, the CBS and CSP modules efficiently extract image features but contribute to a high number of model parameters, resulting in increased computational demands. Addressing this, the upgraded backbone replaces CBS and CSP with GhostConv and C3Ghost modules, effectively decreasing model parameters and lowering computational complexity. Prioritizing the CA module over the C3Ghost module enhances accuracy by suppressing unnecessary data and emphasizing crucial information, minimizing model accuracy loss.

In the neck network's feature fusion process, fewer model parameters impact the speed at which features combine. To address this, the Depthwise Separable Dilated Convolutions (DSDCs) in the neck network combine dilated and depthwise convolutions for a larger receptive field without sacrificing accuracy. Dilated convolutions expand the receptive field without additional parameters, and DSDC further reduces parameters by factorizing the convolution operation. While dilated convolutions can be computationally expensive and DWConv less accurate, the DSDC module combines their advantages, mitigating their drawbacks.

The DSDC module (Zhang & Xian, 2021), a technique replacing the CBS module, considers feature fusion characteristics. Unlike conventional convolution, each DSDC module convolution core handles

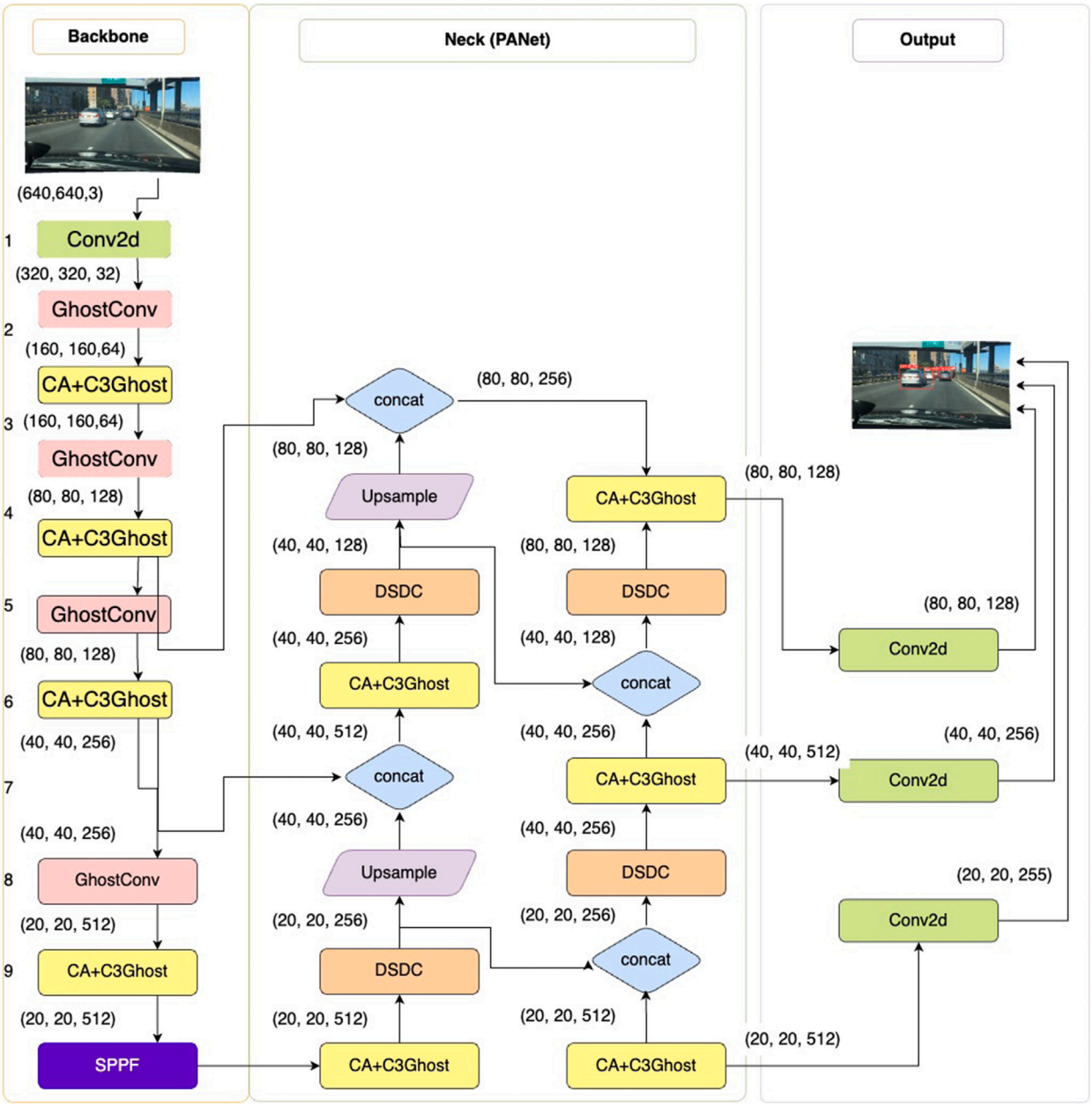



**Fig. 4.** Proposed SD-YOLO-AWDNet model Architecture.

a single channel and acts simultaneously on every channel in a two-dimensional plane, reducing the computational burden. This design, featuring as many convolution cores as channels in the preceding layer (Liu, Hu, Chen, Guo, & Ni, 2023), retrieves local detailed features with fewer parameters.

To strike a balance between multi-scale target detection and accuracy loss, the paper proposes a redesigned structure that combines dilated convolution and DWConv in the shallow feature extractor of the Backbone. This structure overcomes the computational cost increase of dilated convolutions without suffering from the grid effect. It extracts more target features, compensating for the accuracy loss incurred during the process of shrinking and enlarging feature maps.

After each DSDC convolution, the CA+C3Ghost module minimizes parameters and boosts model accuracy.

In the context of adverse weather scenarios for self-driving cars, the YOLOv5 model incorporating GhostConv and DSDC demonstrates proficiency. The SD-YOLO-AWDNet model presents benefits in self-driving cars under adverse weather conditions, encompassing reduced parameters, increased efficiency, enhanced accuracy in challenging situations, and the capability to identify objects of varying scales.

The subsequent subsections illustrate all four levels of enhancements, encompassing GhostConv and C3Ghost, Depth-wise Separable Dilated Convolutions and Residual, CA+C3Ghost, and the Focal-Distribution Loss function.

### 3.2.2. *GhostConv and C3Ghost*

The GhostConv module and the C3Ghost module, as introduced by Ghost-net (Han et al., 2020), were suggested to significantly decrease the parameter count in YOLOv5s. The GhostConv module employs distributed feature map extraction to eliminate redundancy. It employs cheaper linear operations to produce a subset of feature maps and the rest are produced by using Conv modules.

The Ghost-Conv module comprises three distinct stages, namely Convolution, ghost creation, and feature map stitching. In the initial

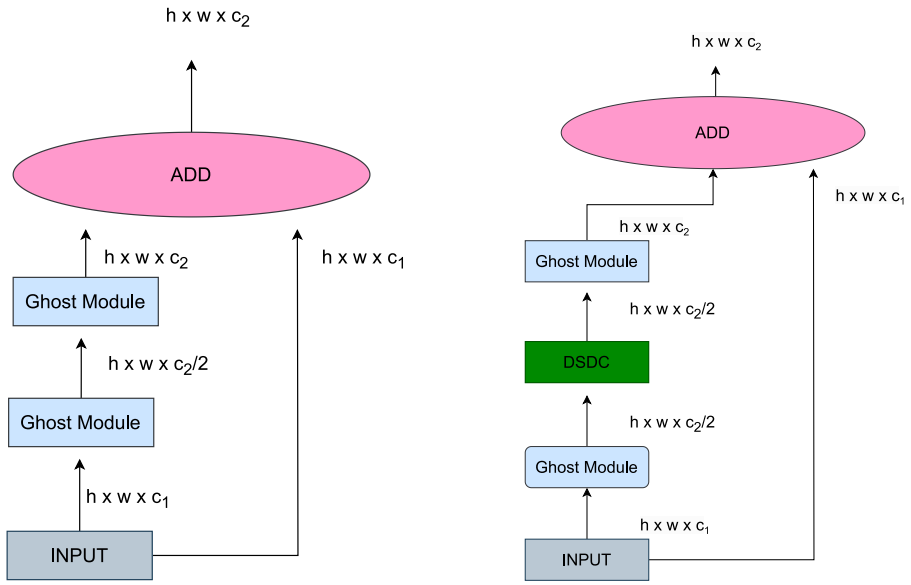



**Fig. 5.** Architectural breakdown of GhostConv and GhostBottleneck modules.

stage, output feature maps are produced by employing a reduced number of convolution kernels. Specifically, the parameter "p" is replaced with a reduced value denoted as "q."

Subsequently, the second phase involves the application of a computationally cheap operation to each feature map generated in the initial step. This results in the creation of a total of $q \times r$ new feature maps, with "r" serving as a factor to ensure that $q \times r$ equals the original parameter "p." This verification ensures the congruence of the feature shapes output by the Ghost module and regular convolution.

The final stage encompasses the combining of these newly generated feature maps, culminating in the completion of the three-stage process of the Ghost-Conv module. Notably, the values of "p," "q," and "r" are integral to the methodology, with "p" representing the original parameter, "q" denoting the reduced value, and "r" ensuring the appropriate relationship between the two for consistency in feature shapes.

Figs. 5(a) and 5(b) depicts the GhostConv module's operating principle.

GhostConv eliminates duplicity through distributed feature map extraction. If each original feature corresponds to "RF" redundant features, GhostConv only needs to generate N/RF base features. It employs linear transforms to expand the original features and create similar ones.

GhostConv's objective is to minimize redundancy in feature maps within conventional convolutions. This involves generating a smaller set of intrinsic features and strategically expanding them, resulting in parameter reduction. The approach includes generating fewer initial feature maps and using cost-effective expansion operations.

The Eqs. (1)–(6) describe the number of FLOPs for both Conventional Convolution and Ghost Convolution, along with FLOPs reduction (Rate_S) and the Parameter compression (Rate_C) of the GhostConv modules.

$$C\_Conv\_Flops = N \times H \times W \times C \times K \times K \tag{1}$$

where, *C_Conv_Flops* represents flops for conventional convolution, $N$ represents the number of feature maps produced by the convolution layer, $H$ represents the height of the Output layer, $W$ represents the width of the Output layer, $C$ represents the number of Input channels, and $K \times K$ represents the size of the convolution kernel and the Flops measure the cost of the convolution layer.

$$GhostConv\_Flops = Flops\_intrinsic + Flops\_ghost \tag{2}$$

where,

$$Flops\_intrinsic = \frac{N}{RF} * H \times W \times C \times k \times k \tag{3}$$

where, *RF* represents the Redundancy factor represents the number of redundant factors.

$$Flops\_ghost = \frac{N}{RF} \times H \times W \times C \times d \times d \times (RF - 1) \tag{4}$$

where, $d \times d$ represents the size of the kernel used for the cheap operations (usually $1 \times 1$).

FLOPs reduction (*Rate_S*) measures how much GhostConv reduces computational cost compared to conventional convolution. As shown in Eq. (5) GhostConv achieves an *RF* times reduction in FLOPs, making it computationally more efficient.

$$Rate\_S = \frac{GhostConv\_Flops}{Flops\_ghost} = RF \tag{5}$$

Parameter compression (*Rate_C*) measures how much GhostConv compresses model parameters compared to conventional convolution. As shown in Eq. (6) GhostConv also reduces the number of parameters by a factor of *RF*, leading to a smaller memory footprint.

$$Rate\_C = Rate\_S = RF \tag{6}$$

The reduction in Flops and parameter compression in GhostConv is directly proportional to the redundancy factor (*RF*). Hence, the choice of *RF* is pivotal in achieving a balance between efficiency and accuracy, as *RF* controls the compression degree. The FLOPs reduction (*Rate_S*) and the parameter compression rate (*Rate_C*) are associated with *RF*, necessitating a trade-off between efficiency and accuracy. GhostConv is particularly beneficial for memory-constrained devices or tasks prioritizing model size.

### 3.2.3. Depth-wise separable dilated convolutions and residual

The integration of depth-wise separable convolutions (Chollet, 2017), dilated convolutions (Yu & Koltun, 2015), and residual connections (Wei, Yuan, Shen, & Zhang, 2017) enhances computational efficiency, improves the capacity to capture large-scale spatial relationships, and facilitates smoother information flow, addressing challenges such as vanishing gradients in deeper learning.

Depth-wise separable dilated convolutions (DSDCs) capitalize on the advantages of depth-wise separable convolutions, dilated convolutions, and residual connections, resulting in the creation of an efficient network.

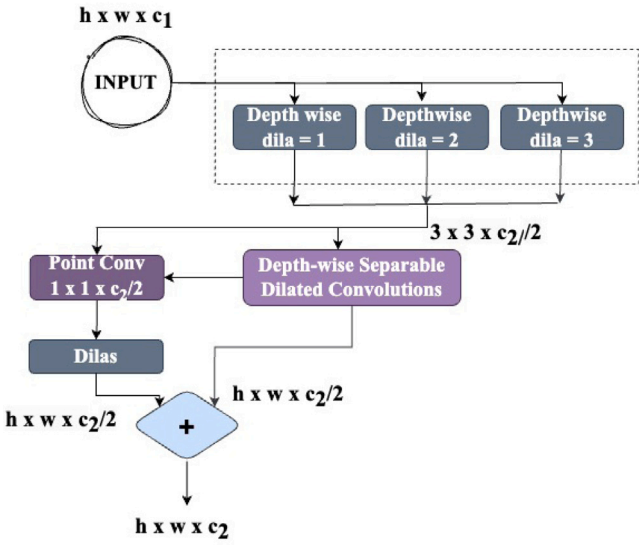



**Fig. 6.** Structure of Depthwise Separable Dilated Convolutions module.

In YOLOv5's approach to object detection, a trade-off is employed, necessitating down-sampling through convolution operations to compress the feature map and enhance the receptive field. This is followed by up-sampling to restore the image size for recognizing targets of diverse scales. However, this can lead to information loss, potentially hindering the detection of certain targets. So, DSDCs offer a potential solution by combining point-wise convolution, dilation, and depth-wise convolution:

- Depth-wise convolution: Separate filters process individual input channels independently, reducing computations.
- Dilation: Gaps in the filters expand the receptive field, capturing long-range dependencies without losing spatial resolution.
- Point-wise convolution: $1 \times 1$ filters merge features and manage output channels, maintaining sensitivity to targets of different sizes.

This approach balances efficiency, receptive field size, and multi-scale feature detection for object detection tasks. The comprehensive architecture of depth-wise separable convolutions is illustrated in Fig. 6.

The input goes through a series of depth-wise separable convolutional layers. Each layer includes two stages:

- Depth-wise convolution: This stage uses multiple filters, each with the same size as a single channel in the input. These filters convolve with each input channel independently, extracting spatial features specific to each channel. This reduces the number of computations compared to standard convolution.
- Point-wise convolution ($1 \times 1 \times N$): This stage uses a single $1 \times 1$ filter for each output channel. This filter combines the extracted features from the depth-wise convolution stage and controls the number of output channels.

Depth-wise separable convolution layers use dilation. This means that the filters "hop" over pixels instead of sliding smoothly across them. This increases the receptive field (area covered) of the filters without making them larger, enabling the network to grasp long-range dependencies in the input.

Residual Connections (Wei et al., 2017) are the skip connections that are added between some of the layers in the network. These connections bypass some layers and directly add their output to the output of later layers. This helps alleviate the vanishing gradient problem and improves information flow for multi-scale target detection, allowing the network to learn deeper representations.

### 3.2.4. CA+C3Ghost module

The Coordinated Attention (CA) module (Hou et al., 2021) enhances attention focus by incorporating positional information. By utilizing two pooling layers, it aggregates information in two distinct spatial orientations and segregates them into one-dimensional features. Recording location data in one spatial direction and monitoring remote dependencies in the opposite direction, the module encodes direction-aware and location-aware attention maps. Utilizing the resultant feature maps for the input amplifies the extraction of meaningful features.

As depicted in Fig. 7, the CA module dynamically adjusts attention across spatial dimensions, enabling the model to efficiently concentrate on pertinent regions and enhance feature representation.

The C3Ghost module, aimed at replacing CBS and CSP in YOLOv5, efficiently extracts features while reducing model parameters for enhanced computational efficiency. Illustrated in Fig. 8, the module incorporates GhostConv layers, collaboratively capturing and processing crucial features from input data. Highlighting the module's design and internal connections, the C3Ghost module efficiently uses parameters. It ensures effective feature extraction while minimizing parameters to optimize resource usage and enhance task performance.

To further improve attention focus on positional information, the CA+C3Ghost module reduces the number of parameters while preserving accuracy. This design aims to emphasize task-relevant features, suppress unnecessary ones, and enhance the efficiency and accuracy of processing image feature information.

Fig. 9 demonstrates the seamless integration of the CA module into the existing C3Ghost structure, showcasing the collaborative functionality of both components.

The CA block captures low-level features from the input image through stacked convolutional layers. The C3Ghost block combines regular convolution for primary feature extraction, "Ghost channels" to cut down computation while preserving accuracy, and extra convolutions for fine-tuning. Finally, it outputs concatenated features from both blocks for a more comprehensive representation.

Overall, CA+C3Ghost efficiently extracts multi-level features from inputs. The C3Ghost module emphasizes parameter efficiency, and the CA+C3Ghost module extends this efficiency by integrating spatial attention mechanisms through the CA module.

### 3.3. SD-YOLO-AWDNet with focal-distribution loss

Within the object detection domain, bounding box regression serves as a pivotal component in determining the accuracy of object localization. YOLOv5 object detection model employs Complete Intersection over Union (CIoU) Loss as the designated function for regression of bounding boxes. CIoU, which is an extension of Distance-intersection over Union (DIoU), induces an increase in loss in proportion to the dimensions of the predicted box. Precisely, it intensifies the penalty linked to the length-width ratio value between the predicted and the ground-truth box as shown in Eq. (7), and Eq. (8):

$$1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \left(\frac{V}{(1 - IoU) + V}\right) V \tag{7}$$

$$V = \frac{4}{\pi^2}\left(arctangent\frac{w^{gt}}{h^{gt}} - arctangent\frac{w}{h}\right)^2 \tag{8}$$

'$b$' and '$b^{gt}$' signify the centre for the predicated and the actual box, respectively. Meanwhile, '$\rho$' refers to the Euclidean distance b/w the centre for the ground truth and the predicted box. '$c$' represents the diagonal length of the minimum bounding rectangle that encloses both the predicted and ground truth bounding boxes. The height and width for the ground truth bounding box are represented by variables '$h^{gt}$' and '$w^{gt}$', while '$h$' and '$w$' represent the height and width for the predicted bounding box predicted. '*Arctangent*' is an Inverse tangent function, returning the angle whose tangent is a given value. Applying the *arctangent* function to the calculated aspect ratio results in an angle between −90 and 90 degrees, representing the "skewness" of the box.

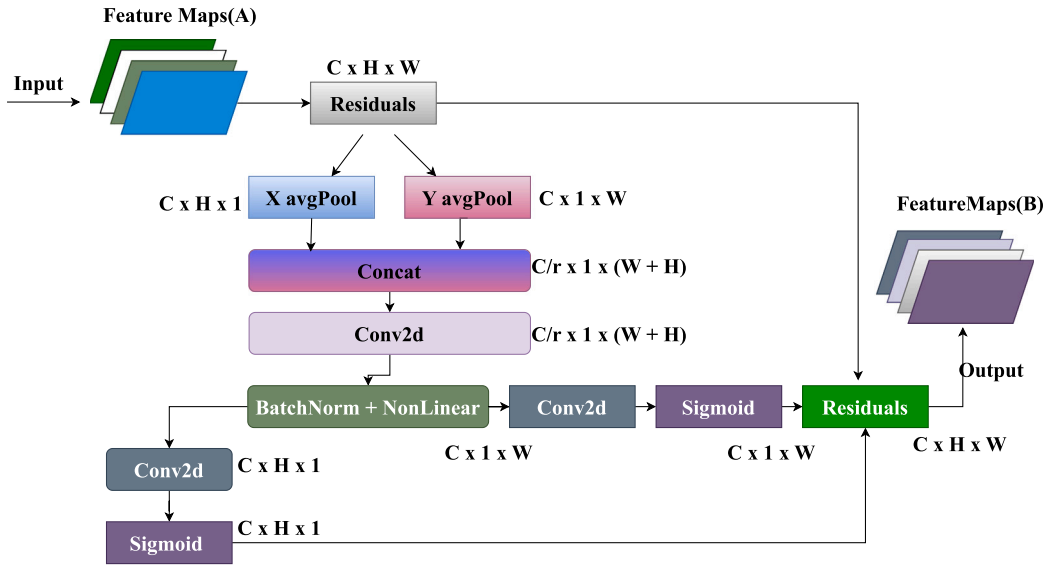



**Fig. 7.** An illustration of coordinated attention module.

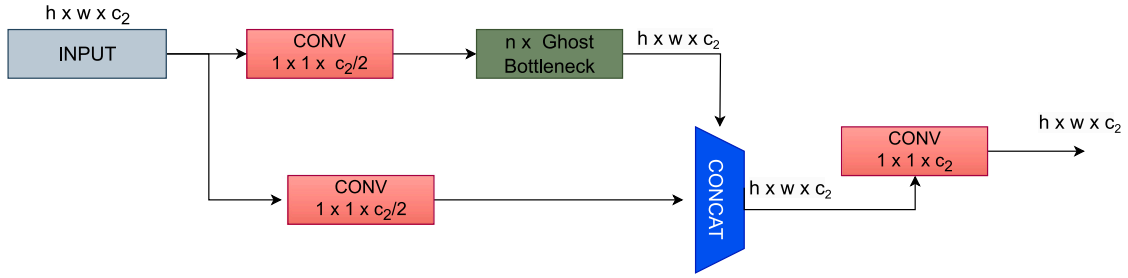



**Fig. 8.** Visual depiction of C3 Ghost module architecture with pointwise convolutions, depthwise convolutions, and linear bottleneck.

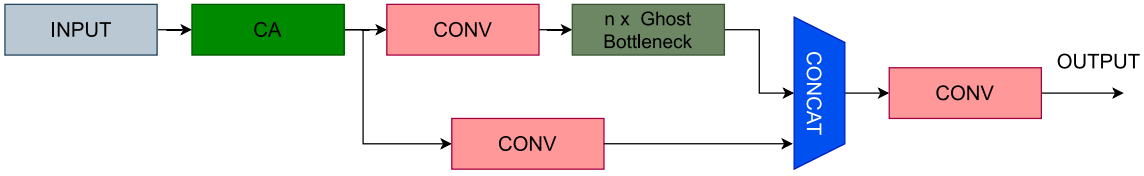



**Fig. 9.** Visualizing the CA+C3Ghost Module.

The CIoU evaluates the similarity between predicted and ground-truth bounding boxes, considering their aspect ratios. It mandates independent adjustments to the height and width of the predicted box to preserve a balance between challenging and straightforward samples. However, the CIoU lacks consideration for maintaining a balance between difficult and easy samples when resizing the boxes.

Efficient-Intersection over Union (EIoU), a metric proposed in Zhang et al. (2022), extends upon the CIoU metric by incorporating a separate measure of the length-to-width ratio. EIoU considers three specified geometric factors (overlapping area, centre point, and side lengths) when assessing the similarity between ground truth and predicted bounding boxes. This enables a more comprehensive and accurate measurement of the disparity between the ground truth and predicted boxes.

There can be issues while using imbalanced training data, like when dealing with samples that are inherently more difficult to classify than others. The Focal Loss function was proposed as a solution for addressing this issue. By modifying the standard cross-entropy loss function, Focal Loss assigns a higher weight to misclassified samples that are more difficult to classify, thus giving more attention to those samples during training. This approach has demonstrated enhanced performance in models addressing imbalanced classification tasks.

Eqs. (9) and (10) provide the calculation method for the above technique:

$$L_{EIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \frac{\rho^2(w, w^{gt})}{c_w^2} + \frac{\rho^2(h, h^{gt})}{c_h^2} \tag{9}$$

$$L_{FocalEIoU} = IoU^{\gamma} L_{EIoU} \tag{10}$$

The proposed focal-distribution loss function inherits the advantages of FocalEIoU loss while intensifying loss for high Intersection over Union (IoU) targets, ensuring more precise localization. Additionally, it improves gradient-adaptive weighted regression for bounding boxes, leading to more accurate object detection. This addresses accuracy degradation caused by occlusion and perspective distortion issues, even in diverse weather conditions. The calculation method for it is given in Eq. (11):

$$\begin{aligned} L_{Focal-DistributionLoss} = & IoU^{\gamma} \left( (\alpha(1 - IoU)^{\beta}(1 - IoU)^{\delta}) \right. \\ & \left. + \max(0, R(b) + R(w) + R(h)) \right) \end{aligned} \tag{11}$$

**Table 1**
Number of objects in train and test dataset in BDD100k.

| | Traffic sign | Traffic light | Car | Rider | Motor | Drivable area | Lane | Person | Bus | Truck | Bike | Train |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Training data | 2,339,686 | 1,86,117 | 4,517 | 3,002 | 1,25,723 | 5,28,643 | 91,349 | 11,672 | 29,971 | 7,210 | 136 | 122 |
| Validation data | 34,908 | 26,885 | 1,02,506 | 649 | 452 | 17,981 | 75,730 | 13,272 | 1,597 | 4,245 | 1,007 | 15 |

**Table 2**
Number of images per weather.

| | Clear | Rainy | Snowy | Overcast | Partly cloudy | Foggy | Unknown |
|---|---|---|---|---|---|---|---|
| Training data | 37,344 | 5,070 | 5,549 | 8,770 | 4,881 | 130 | 8,119 |
| Validation data | 5,346 | 738 | 769 | 1,239 | 738 | 13 | 1,157 |

**Table 3**
Number of images per scene.

| | City street | Highway | Residential | Parking lot | Tunnel | Gas stations | Unknown |
|---|---|---|---|---|---|---|---|
| Training data | 43,516 | 17,379 | 8,074 | 377 | 129 | 27 | 361 |
| Validation data | 6,112 | 2,499 | 1,253 | 49 | 27 | 7 | 53 |

**Table 4**
Number of images per time of day.

| | Daytime | Dawn/Dusk | Night | Unknown |
|---|---|---|---|---|
| Training data | 36,728 | 5,027 | 27,971 | 137 |
| Validation data | 5,258 | 778 | 3,929 | 35 |

$$R(b) = \frac{\rho^{2\alpha}(b, b^{gt})}{c^2} \tag{12}$$

$$R(w) = \frac{\rho^{2\beta}(w, w^{gt})}{c_w^2} \tag{13}$$

$$R(h) = \frac{\rho^{2\beta}(h, h^{gt})}{c_h^2} \tag{14}$$

The Focal-Distribution Loss function is structured to address challenges such as imbalanced class distributions by introducing a focal term and a regularization term.

The term $(\alpha(1 - IoU)^{\beta}(1 - IoU)^{\delta})$ introduces a focal term that dynamically adjusts the loss based on the IoU.

The regularization term *max(0, R(b) + R(w) + R(h))* penalizes bounding box regression errors, where, *R(b)*, *R(w)*, and *R(h)* are regularization terms.

The Bounding Box Regularization *R(b)*, Width Regularization *R(w)* and Height Regularization *R(h)* are shown in Eqs. (12), (13), and (14).

Other terms are described as:

$IoU$ represents the actual IoU score between the predicted and actual bounding boxes.

$\gamma$ represents a hyperparameter regulating the significance of the IoU term.

$\rho$ represents a function computing the Euclidean distance between two points.

$\alpha$, $\beta$ represent hyperparameters that control the importance of width, height, and the centre point terms.

$c$, $c_w$, $c_h$ represent hyperparameters that govern the impact of the centre point, width, and height terms.

This paper conducted multiple experiments on the BDD100K dataset to fine-tune our model and subsequently opted for the following hyperparameter values: $\alpha = 2.0$, $\beta = 0.25$ and $\gamma = 2.0$.

## 4. Results

In this section, the dataset, training environment, and performance metrics are described. Following that, multiple experiments are conducted to demonstrate the superior performance of the proposed SD-YOLO-AWDNet model.

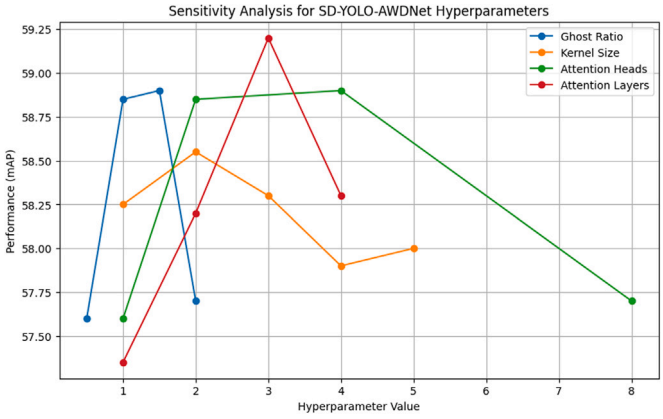



**Fig. 10.** Sensitivity analysis for key hyperparameters for the proposed model.

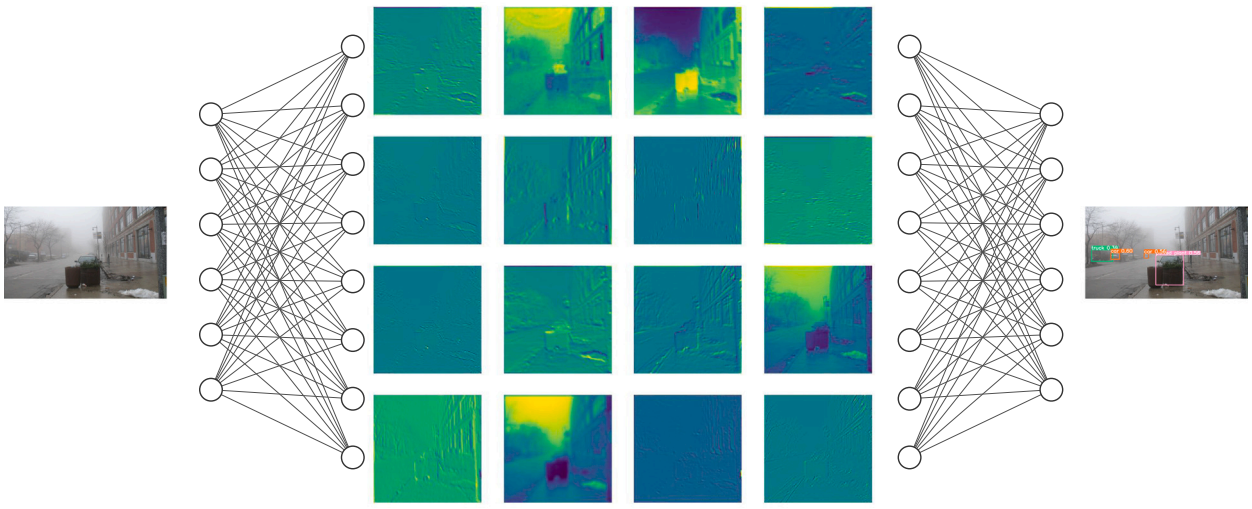
### 4.1. Dataset and computational details

This paper utilizes the BDD100K dataset (Yu et al., 2020), which encompasses diverse scenarios such as highways, residential areas, and city streets. The dataset consists of videos recorded at different periods during the day and under varying weather conditions, distributed across training (70K), validation (10K), and testing (20K) sets. Notably, the dataset features scenes from around the world, with a balanced representation of day and night footage. This diversity facilitates domain transfer, enabling successful generalization of object detection models to new test sets. The dataset includes detailed distributions of images categorized by scene, time of day, and weather, as depicted in the Tables 1–4. To focus more on out-of-distribution scenarios, the BDD100k dataset has been divided into 6 different categories that handle the corner cases covering adverse scenarios of clear, overcast, foggy, partly cloudy, rainy, and snowy. BDD100k includes bounding box annotations for 10 detection categories in the reference frames. For validation purposes, the proposed model is also evaluated on datasets of CADC (Pitropov et al., 2021) and KITTI (Geiger et al., 2013). The sample images from Indian road traffic are also used for performance validation, collected across different places and weather scenarios.

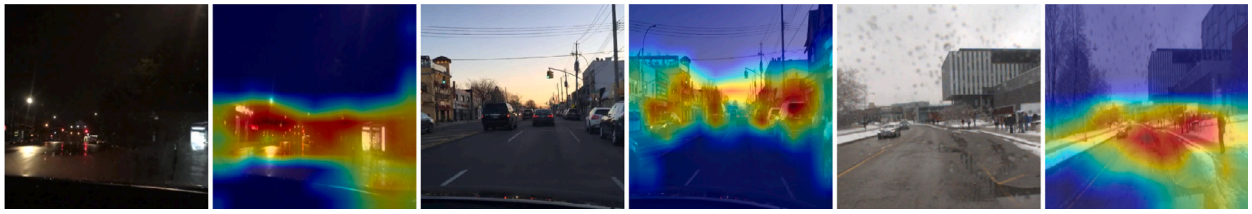
Table 5 presents the system configuration of the computing setup, showcasing key components. The system comprises 32 GB of total memory, an Nvidia Tesla T4 GPU with 16 GB memory, and an Intel(R)

**Table 5**
System configuration.

| Total memory | GPU | CPU | Cache size | CUDA Version | CudNN |
|---|---|---|---|---|---|
| 32 GB | Nvidia Tesla T4 (16 GB) | Intel(R) Xeon(R) @ 2.30 GHz * 4 core | 46080 KB | 11.2 | 8.1.0.77-1 |



((a)) High-level graphical representations of the internal workings of the object detection process



((b)) Heap maps generated by SD-YOLO-AWDNet after applying CA mechanism

**Fig. 11.** Overview of the object detection process and results.

Xeon(R) CPU operating with 4 cores at 2.30 GHz. Additionally, the details on the cache size (46080 KB), CUDA (11.2), and CudNN (8.1.0.77-1) versions are also provided in the table offering a comprehensive overview of the hardware and software specifications.

Default hyper-parameters values for YOLOv5 model is slower to converge. So, this paper performs a sensitive analysis for the additional hyper-parameters (Ghost module ratio, kernel sizes, and attention mechanism parameters, added by the Ghost Network in the proposed model. Fig. 10 represents the output of sensitivity analysis. The $x$-axis represents impact values corresponding to each hyper-parameter and the $y$-axis measures the performance of the model using the mean Average Precision (mAP) metric. Based on the sensitivity analysis, the impact values for hyperparameters are deduced.

### 4.2. Evaluation metrics

The model's performance is assessed using various technical metrics, including Precision, Recall, mean Average Precision (mAP), Frames Per Second (FPS), the number of parameters, Floating Point Operations per Second (FLOPs), and loss function curves. All these metrics are utilized for quantitatively assessing the efficacy and efficiency of the examined model. All the metrics are outlined in the following sections.

The Precision (P) metric in classification estimates the accuracy of positive predictions, representing the proportion of correctly predicted positive instances among all instances predicted as positive. It is computed as the ratio of true positives to the sum of true positives and false positives, as illustrated in the Eq. (15):

$$P = \frac{TPvalues}{TPvalues + FPvalues} \tag{15}$$

Recall ($R$) quantifies the ratio of correctly anticipated positive cases among all positive instances, as given in Eq. (16):

$$R = \frac{TPvalues}{TPvalues + FNvalues} \tag{16}$$

In Eqs. (15) and (16), $FPvalues$ indicate the number of detections found solely in the results, not in the ground truth. $TPvalues$ denote the number of detections present in both the ground truth and the results. $FNvalues$ represent the number of detections existing in the ground truth only, absent in the results.

Recall and Precision are two commonly used performance metrics in machine learning that are often in conflict with each other. To address this, the $mAP$ is introduced, which combines both Precision and Recall to provide an overall score across all classes that reflects the algorithm performance. It calculates the average precision across various recall levels, providing a comprehensive evaluation of a model's ability to precisely identify objects within a given context. The algorithm's performance improves with a higher $mAP$ score. It is calculated as shown in Eq. (17):

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i \tag{17}$$

Here:

$N$ represents the total count of classes or categories and $AP_i$ represents the Average Precision for the class.

For a specific class, the Average Precision is given in Eq. (18) and calculated as:

$$\mathrm{AP}_i = \frac{1}{R_i} \sum_{k=1}^{R_i} P(k) \cdot precAtK(k) \tag{18}$$

where:

$R_i$ is the overall count of relevant instances in class.

$P(k)$ is the precision for $k$th retrieved instance.

$precAtK(k)$ is an indicator function that equals *1* if the $k$th retrieved instance is relevant, and *0* otherwise.

Frames Per Second is a measure of a system's or model's processing speed. It represents the number of image frames processed or generated per second, reflecting the system's real-time processing capability.

The number of Parameters refers to the biases and weights in the model. It reflects the model complexity and its ability to learn from the training data, with a higher number of parameters often indicating increased model complexity.

Floating-point operations per second (FLOPs) is a metric that quantifies the computational complexity of a model, representing performed floating-point operations per second. It is a crucial measure for understanding the computational efficiency and performance of a model, especially in resource-constrained environments.

### 4.3. Experimental graphs and detection results on BDD100k dataset

In this section, the detection results are listed for the proposed SD-YOLO-AWDNet. Later, the performance of the proposed SD-YOLO-AWDNet is presented in contrast to YOLOv5s using various performance metrics. Finally, the performance of SD-YOLO-AWDNet is compared with other state-of-the-art models.

#### 4.3.1. Proposed SD-YOLO-AWDNet detection results

In this section, results for intermediate steps and the model's output are analysed. Fig. 11 provides an overview of the object detection process and results. Fig. 11(a) illustrates an intermediate training state of the input image using the SD-YOLO-AWDNet model, converting the original image into pixels. Through graph networks, feature detection utilizes filters to activate multiple neurons, highlighting pattern recognition within the input image, which is further refined for object identification and classification. Explainable AI techniques are employed to enhance interoperability across different model components and detection layers. Fig. 11(b) depicts heat maps generated by the proposed model post-integration with the CA mechanism, aimed at clarifying the architecture's operational coherence.

The detection outcomes of SD-YOLO-AWDNet are presented in the subsequent Table 6 concerning six different scenarios i.e. clear sky, foggy, night, partly cloudy, rainy and snowy. Detection output showcases bounding boxes for various classes with confidence scores attached to them.

#### 4.3.2. Proposed SD-YOLO-AWDNet vs YOLOv5s

A comparison of the performance of the proposed algorithm with the base model YOLOv5s is discussed in this subsection.

*4.3.2.1. Ablation experiment.* In this section, the ablation experiment results are explained for the proposed SD-YOLO-AWDNet with YOLOv5, YOLOv5-GN and YOLOv5-FDL. For YOLOv5-GN, ghost conv, c3 ghost modules and DSDCs are added in the backbone to make the model lightweight and enhance feature extraction by incorporating the DSDC and CA module in the GhostBottleneck. For YOLOv5-FDL, a novel loss function named Focal Distribution Loss is employed. Subsequent subsections will include mAP, loss and precision comparison curves for comparative analysis with the proposed model.

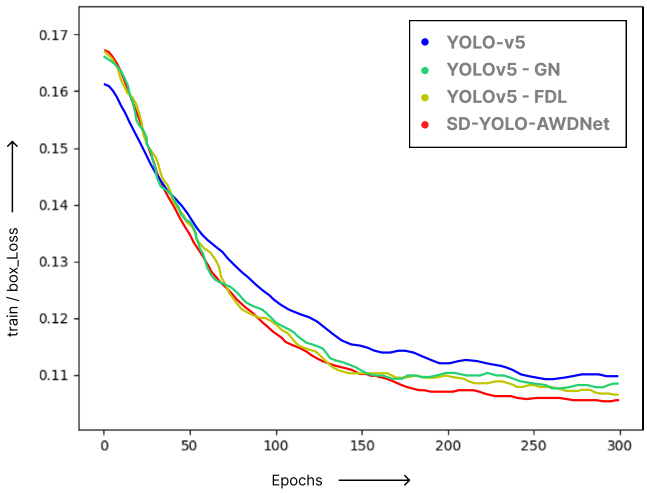



**Fig. 12.** Loss comparison of YOLOv5s and SD-YOLO-AWDNet.

**• mAP Comparison**

The ablation experiments were performed on four different models, including YOLOv5, YOLOv5-GN, YOLOv5-FDL and SD-YOLO-AWDNet, to confirm the efficiency of the proposed model. As shown in Table 7, the number of parameters and FPS are reduced to half in comparison to YOLOv5. A slight decrease in mAP is observed from 57.8 in YOLOv5 to 57.3 in YOLOv5-GN. For YOLOv5-FDL, an increase in the mAP value from 57.8 (in YOLOv5) to 59.2 is observed with the same parameters and FPS. Both YOLOv5-GN and YOLOv5-FDL are combined in SD-YOLO-AWDNet to provide a light-weight model with fewer parameters (half of YOLOv5) and better mAP values (59.1) providing an enhancement of 2.24% as per the ablation experiments.

**• Loss Comparison Curve**

Loss function curves illustrate the variation of the model's loss over training epochs. The loss function measures the disparity between the predicted and actual values, and the curve offers insights into the model's learning process. Monitoring loss function curves helps in assessing convergence, identifying over-fitting or under-fitting, and optimizing model performance. Fig. 12 represents the bounding box loss of two models named YOLOv5s and the proposed SD-YOLO-AWDNet over training data in 300 epochs during training. A faster decline for SD-YOLO-AWDNet represents more efficient learning than YOLOv5s. Major improvements over the YOLOv5 model are also described as YOLOv5 - GN (Ghost Network) and YOLOv5 - FDL (Focal Distribution Loss), showcasing how these intermediate improvements impact change in loss for training on the BDD100k dataset.

**• Precision Comparison Curve**

The precision curve comparison between YOLOv5 and the proposed SD-YOLO-AWDNet reveals intriguing insights into their respective performances. Precision, a critical metric in object detection, measures the accuracy of identified objects. On analysis of precision curves for both models, it can be justified that the proposed model consistently performs better.

Fig. 13 shows the precision comparison of YOLOv5s and SD-YOLO-AWDNet over 300 epochs of training data. It is evident that the SD-YOLO-AWDNet results in fewer misclassifications and has better precision in object detection within the scene. Similar to the previous section, intermediate improvements of YOLOv5 - GN (Ghost Network) and YOLOv5 - FDL (Focal Distribution Loss) are also added here to showcase how these improvements impact change in precision curve over training on the BDD100k dataset.

SD-YOLO-AWDNet shows more sensitivity in detecting smaller objects, which is generally a common challenge in object detection tasks. The model performs well for a diverse scale of objects.

**Table 6**
Detection results of proposed SD-YOLO-AWDNet on different weather scenarios.

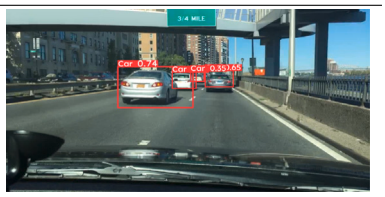
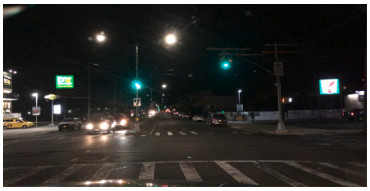
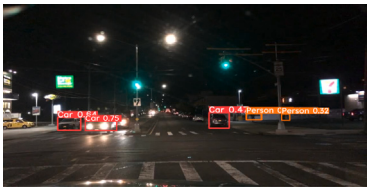
| Weather scenarios | Original image | SD-YOLO-AWDNet Detection |
|---|---|---|
| Clear sky |  |  |
| Foggy |  |  |
| Night |  |  |
| Partly cloudy |  |  |
| Rainy |  |  |
| Snowy |  |  |

**Table 7**
Results from ablation experiments with BDD100k dataset.

| Model | mAP (%) | Precision (%) | Recall (%) | Parameters | FLOPs (G) |
|---|---|---|---|---|---|
| YOLOv5s (Jocher et al., 2022) | 57.8 | 82.12 | 77.0 | 7 046 599 | 16.0 |
| YOLOv5-GN | 57.3 | 82.14 | 69.51 | 3 708 815 | 8.3 |
| YOLOv5-FDL | 59.2 | 90.34 | 78.0 | 7 046 599 | 16.0 |
| **SD-YOLO-AWDNet** | **59.1** | **1.0** | **0.78** | **3 708 815** | **8.3** |

**Table 8**
mAP (%) comparison of YOLOv5s and SD-YOLO-AWDNet on different weather scenarios of BDD100k.

| Model | Clear sky | Foggy | Night | Partly cloudy | Rainy | Snowy |
|---|---|---|---|---|---|---|
| YOLOv4 (Bochkovskiy, Wang, & Liao, 2020) | 54.23 | 47.17 | 48.19 | 48.01 | 46.23 | 47.58 |
| YOLOv5s (Jocher et al., 2022) | 54.31 | 51.65 | 54.01 | 52.34 | 51.12 | 52.28 |
| **SD-YOLO–AWDNet** | **62.80** | **56.97** | **58.23** | **59.81** | **54.32** | **53.89** |

**Table 9**
mAP comparison of YOLOv5s and SD-YOLO-AWDNet on different classes of BDD100k.

| Class | YOLOv5s (Jocher et al., 2022) | SD-YOLO-AWDNet |
|---|---|---|
| Bike | 0.550 | 0.550 |
| Bus | 0.635 | 0.642 |
| Car | 0.832 | 0.845 |
| Motor | 0.482 | 0.493 |
| Person | 0.687 | 0.671 |
| Rider | 0.478 | 0.479 |
| Traffic light | 0.708 | 0.768 |
| Traffic sign | 0.751 | 0.782 |
| Train | 0.000 | 0.021 |
| Truck | 0.660 | 0.662 |
| **mAP** | **0.578** | **0.591** |

The prevision curve for SD-YOLO-AWDNet also has a smoother ascent, indicating a more stable and reliable performance. As compared to YOLOv5's inconsistent behaviour for some classes, SD-YOLO-AWDNet shows high precision values across diverse adverse scenarios with consistency across all classes.

In short, the precision curve comparison between SD-YOLO-AWDNet and YOLOv5 showcases the superior performance of SD-YOLO-AWDNet with fewer false positives, consistency across all classes and handling diverse scales of objects. SD-YOLO-AWDNet outperforms YOLOv5 in terms of model accuracy and reliability.

*4.3.2.2. Class-wise comparative analysis.* In this section, a comparative analysis is done between SD-YOLO-AWDNet with YOLOv5 based on different categories of the BDD100k dataset. The following sections will include class-wise comparison based on mAP values for BDD100k object classes, comparison based on performance on individual adverse weather conditions and PR curve comparison based on BDD100k object classes.

**• Class-wise mAP Comparasion**

Table 8 presents the mAP values for YOLOv5 and SD-YOLO-AWDNet across various weather scenarios. Furthermore, YOLOv4 is incorporated in the table due to its comparable performance with the YOLOv5s model, and in certain classes (for example, clear sky), it even outperforms YOLOv5s. The result showcases better mAP values in all scenarios for the proposed model of SD-YOLO-AWDNet.

**• Adverse Weather Based mAP Comparasion**

In Table 9, classwise mAP comparisons are also listed for better clarity of results. It represents mAP comparisons of YOLOv5s and SD-YOLO-AWDNet on all eight classes of the BDD100k dataset,i.e, it is evident that SD-YOLO-AWDNet performs better than YOLOv5 in most of the classes and achieves the highest overall class based mAP scores.

**• The Precision-Recall (PR) Curve**

The precision–recall (PR) curve presents the relationship between the metrics of precision and recall across various thresholds, providing a clear assessment of the model's performance at different levels of precision and recall.

The precision–recall (PR) curve is graphically depicted, with the $x$-axis denoting the recall metric and the $y$-axis denoting the precision metric. Originating from a point based on the initial threshold, the curve illustrates how precision and recall change as the detection threshold is adjusted. Typically, raising the threshold results in an increase in precision but a decrease in recall. This is because, at higher thresholds, the model becomes more selective and makes detections only when it is more certain, leading to fewer false positives but potentially overlooking some true positives.

The precision–recall curve and mAP values of all classes in the BDD100k dataset for YOLOv5 and SD-YOLO-AWDNet are shown in Figs. 14 and 15. By analysing the PR curves, it can be confirmed that the proposed model exhibits a higher area under the curve, suggesting superior overall performance.

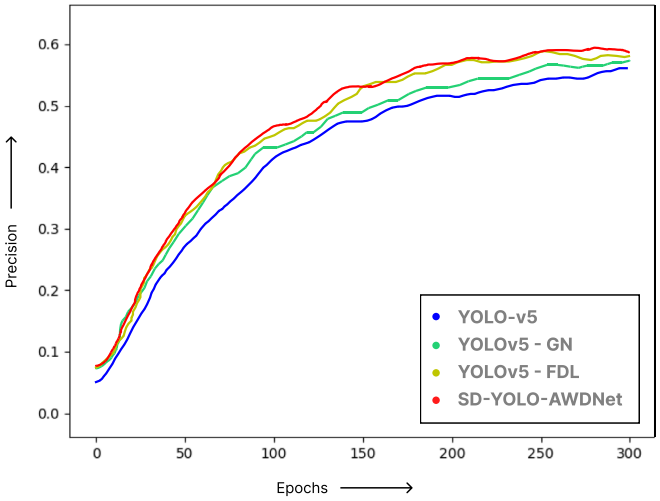



**Fig. 13.** Precision comparison of YOLOv5s and SD-YOLO-AWDNet.

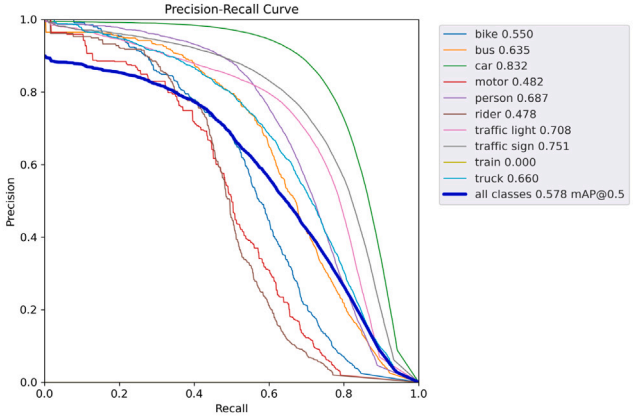



**Fig. 14.** PR curve showing mAP values of YOLOv5s on BDD100k for all classes.

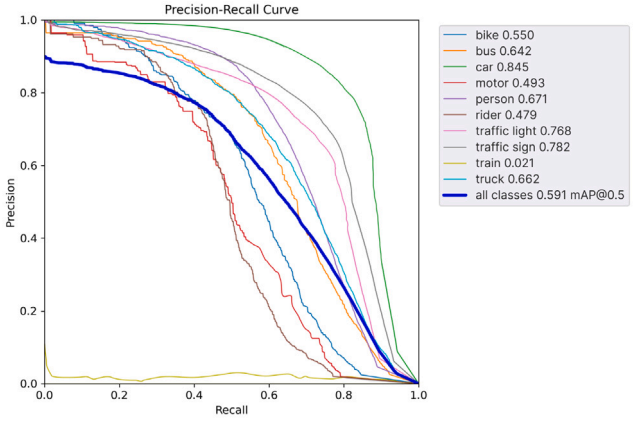



**Fig. 15.** PR curve showing mAP values of SD-YOLO-AWDNet on BDD100k for all classes.

*4.3.3. Proposed SD-YOLO-AWDNet vs SOTA (state of art models)*

Result comparison of the proposed algorithm with State Of Art Models is discussed in this subsection.

**Table 10**
Comparative performance metrics analysis: mAP, FPS, GFLOPs, and parameters across state-of-the-art models on the BDD100k dataset.

| Model | mAP@0.50 (%) | mAp@0.75 (%) | FPS | FLOPs (G) | Parameters |
|---|---|---|---|---|---|
| FasterRCNN (Girshick, 2015) | 29.32 | 14.81 | 27.06 | 231.43 | 42.4 million |
| SSD Resnet (Liu et al., 2016) | 34.31 | 18.23 | 54.93 | 95.86 | 25.6 million |
| CenterNet (Duan et al., 2019) | 41.92 | 28.45 | 42.15 | 63.08 | 21.3 million |
| Yolov4 (Bochkovskiy et al., 2020) | 50.23 | 31.46 | 40.46 | 128.73 | 64.4 million |
| Yolov5s (Jocher et al., 2022) | 57.80 | 37.53 | 56.95 | 16.0 | 7.5 million |
| Yolov6 (Li, Li, et al., 2022) | 55.89 | 42.23 | 142 | 27.7 | 19.6 million |
| Yolov7 (Wang, Bochkovskiy & Liao, 2023) | 56.8 | 29.0 | 161 | 104.3 | 37 million |
| Yolov7s-tiny (Cheng et al., 2023) | 51.1 | 41.7 | 278 | 5.8 | 6.2 million |
| Yolov8s (Vats & Anastasiu, 2023) | 58.1 | 45.9 | 87.1 | 28.6 | 11.2 million |
| **SD-YOLO-AWDNet** | **59.13** | **44.19** | **80.21** | **8.3** | **3.7 million** |

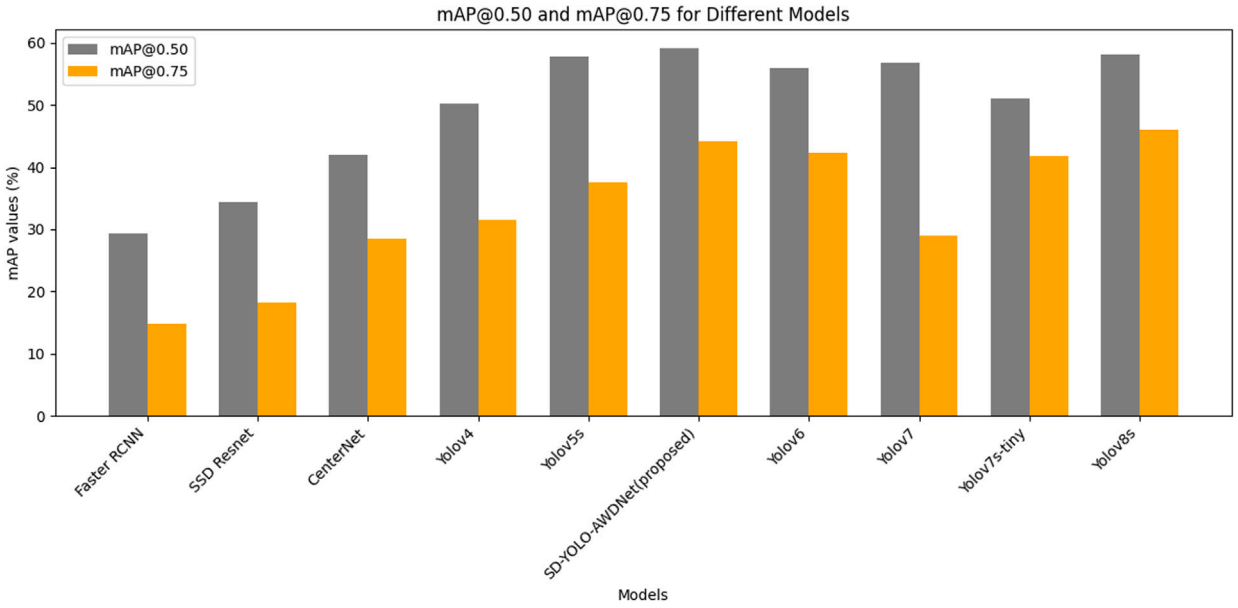



**Fig. 16.** mAP values comparison of SD-YOLO-AWDNet with State-Of-Art-Models.

**• mAP Value Based Comparison**

The comparison Table 10 shows the mAP@0.50, mAP@0.75, FPS, Giga Floating Point Operations Per Second (GFLOPs) and number of parameters values of different object detection algorithms for the BDD100k dataset. It is evident that the proposed model, SD-YOLO-AWDNet, demonstrates superior performance compared to all other models across various performance metrics, as indicated by higher mAP values. The improvement is notable considering the model has nearly half the parameters compared to YOLOv5s. In addition to the improved accuracy, there is a noticeable boost in FPS, signifying a faster processing speed. The model also demonstrates a decrease in GFLOPs, highlighting its efficiency in computational operations. Together, these aspects highlight the lightweight design and heightened performance efficiency of the proposed model SD-YOLO-AWDNet.

SD-YOLO-AWDNet is the most accurate object detection model of all shown in the table, having the highest mAP score of 59.1. Faster RCNN is the least accurate object detection model of all shown in the table with the lowest mAP score of 29.3. SSD Resnet and CenterNet also performed on the lower scale with mAP values of 34.31 and 41.92, respectively. YOLOv5s, YOLOv6, YOLOv7 and YOLOv7-tiny performed slightly weaker based on mAP values of 57.80, 55.89, 56.8, and 51.1, respectively. The total number of parameters utilized is significantly high as well in these models. YOLOv8s performance is similar but again the parameters used are much higher as well. SD-YOLO-AWDNet has used almost one-third of the total number of parameters utilized by YOLOv8s.

The bar chart as shown in Fig. 16, conveys the comparison of mAP scores including mAP@50 and mAP@75 values for all models vs SD-YOLO-AWDNet. SD-YOLO-AWDNet is the most accurate object detection model of the ten shown having the highest mAP@50 score i.e. 59.1. Faster RCNN is the least accurate object detection model of the ten shown because it has the lowest mAP@50 score, 29.3. While YOLOv8s outperforms all others in mAP@75, SD-YOLO-AWDNet is comparatively lighter in weight than YOLOv8s.

**• Detection Results**

After training models, objects are detected in different weather scenarios with ten state-of-the-art models (CenterNet, Faster RCNN, SSD Resnet, YOLOv5s, SD-YOLO-AWDNet model, YOLOv6, YOLOv7, YOLOv7-tiny, and YOLOV8)

Table 13 illustrates the detection results of SD-YOLO-AWDNet in adverse weather conditions across six distinct scenarios of clear sky, foggy, night, partly cloudy, rainy, and snowy, from the BDD100K dataset. The comparative analysis with other models offers insights into the proposed model's robustness and performance variations under challenging weather conditions. It is evident from the output of the models that the miss ratio is comparatively lesser in the proposed model of SD-YOLO-AWDNet. Reviewing the results aids in comparing SD-YOLO-AWDNet with other models, demonstrating its capability to precisely detect and locate objects in various scenarios.

The visual representation of the detection results helps us grasp SD-YOLO-AWDNet's capability to recognize and accurately detect objects in various environmental challenges. By presenting results across multiple weather scenarios, the figures provide a comprehensive evaluation of SD-YOLO-AWDNet's adaptability and efficacy in real-world conditions for autonomous vehicles.

**Table 11**
Comparison of performance metrics of Proposed model with State-of-art models on the CADC Dataset.

| Model | mAP@ 0.50 (%) | mAp@ 0.75 (%) | FPS | FLOPs (G) | Parameters |
|---|---|---|---|---|---|
| Faster RCNN (Girshick, 2015) | 22.60 | 11.60 | 27.06 | 231.43 | 42.4 million |
| SSD Resnet (Liu et al., 2016) | 1.20 | 0.30 | 54.93 | 95.86 | 25.6 million |
| CenterNet (Duan et al., 2019) | 23.80 | 14.60 | 42.15 | 63.08 | 21.3 million |
| Yolov5s (Jocher et al., 2022) | 25.60 | 13.46 | 40.46 | 128.73 | 64.4 million |
| Yolov6 (Li, Li, et al., 2022) | 25.90 | 15.41 | 142 | 27.7 | 19.6 million |
| Yolov7 (Wang, Bochkovskiy & Liao, 2023) | 27.61 | 15.30 | 161 | 104.3 | 37 million |
| Yolov7s-tiny (Cheng et al., 2023) | 27.22 | 14.90 | 278 | 5.8 | 6.2 million |
| Yolov8s (Vats & Anastasiu, 2023) | 28.20 | 15.68 | 87.1 | 28.6 | 11.2 million |
| **SD-YOLO-AWDNet** | **28.80** | **15.60** | **80.21** | **8.3** | **3.7 million** |

**Table 12**
Performance metrics comparison: Proposed model versus state-of-the-art models on KITTI dataset.

| Model | mAP@ 0.50 (%) | mAp@ 0.75 (%) | FPS | FLOPs (G) | Parameters |
|---|---|---|---|---|---|
| Faster RCNN (Girshick, 2015) | 54.5 | 28.4 | 27.06 | 231.43 | 42.4 million |
| SSD Resnet (Liu et al., 2016) | 44.0 | 10.1 | 54.93 | 95.86 | 25.6 million |
| CenterNet (Duan et al., 2019) | 29.2 | 12.7 | 42.15 | 63.08 | 21.3 million |
| Yolov5s (Jocher et al., 2022) | 83.3 | 54.0 | 40.46 | 128.73 | 64.4 million |
| Yolov6 (Li, Li, et al., 2022) | 83.39 | 54.43 | 142 | 27.7 | 19.6 million |
| Yolov7 (Wang, Bochkovskiy & Liao, 2023) | 84.91 | 53.5 | 161 | 104.3 | 37 million |
| Yolov7s-tiny (Cheng et al., 2023) | 84.12 | 52.88 | 278 | 5.8 | 6.2 million |
| Yolov8s (Vats & Anastasiu, 2023) | 84.99 | 55.0 | 87.1 | 28.6 | 11.2 million |
| **SD-YOLO-AWDNet** | **86.0** | **55.1** | **80.21** | **8.3** | **3.7 million** |

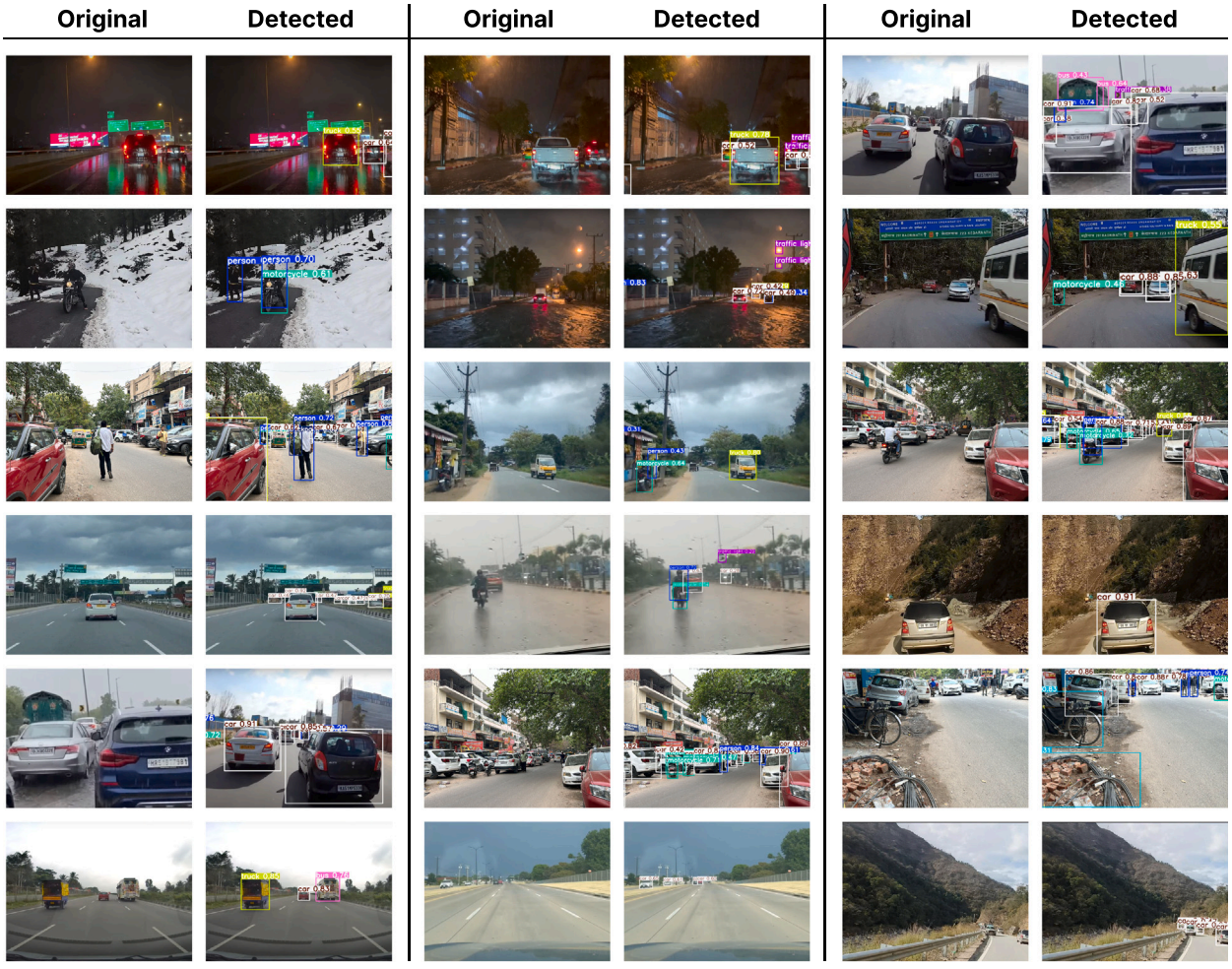
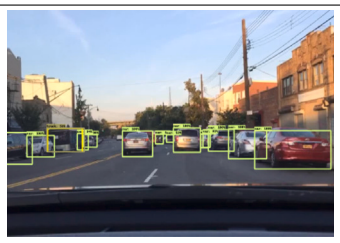
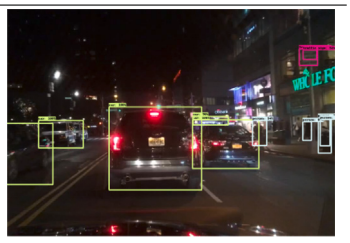
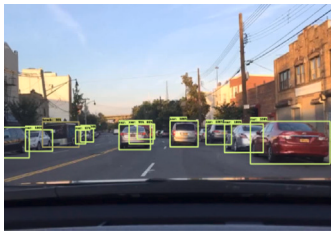
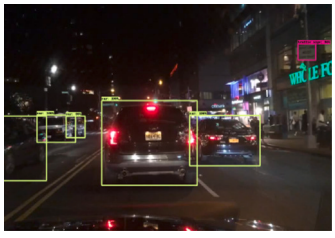
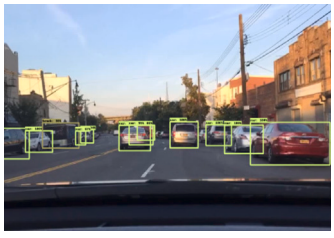
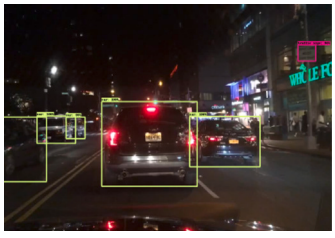
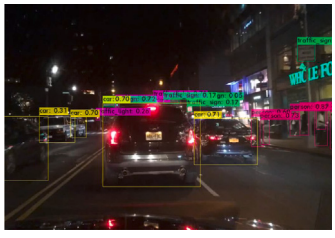
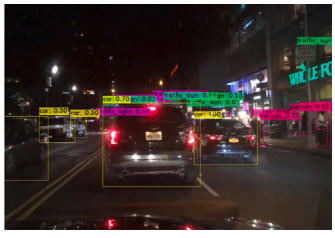
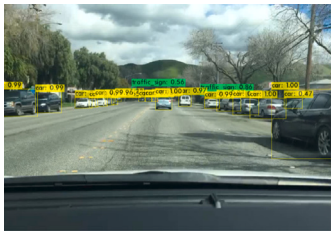


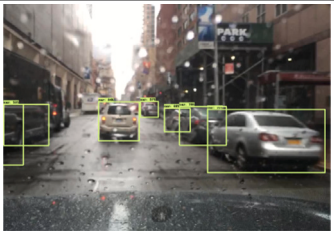
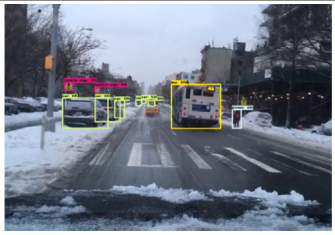
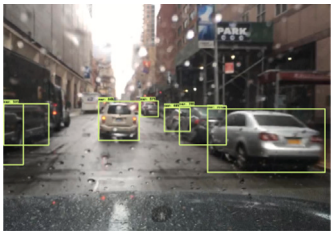
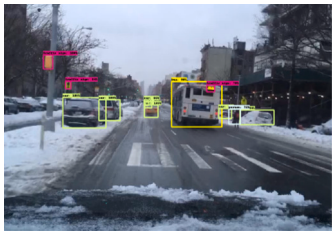
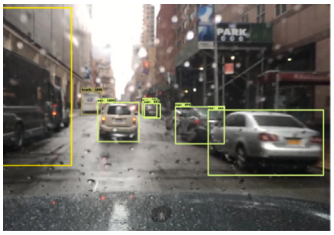
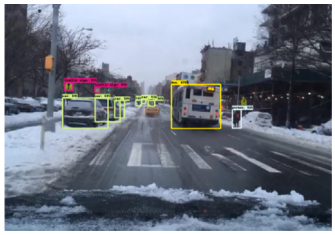
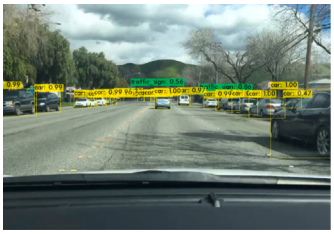
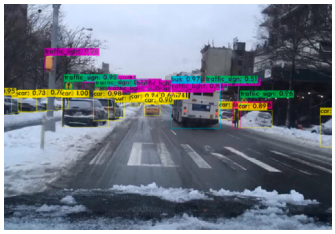
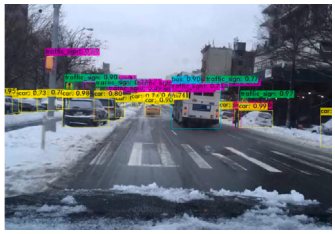

**Fig. 17.** Proposed model's detection results for real world scenarios on different Indian locations.

**Table 13**
Detection results comparison of different models in different weather scenarios with proposed SD-YOLO-AWDNet.

| Models | Clear Sky | Foggy | Night |
|---|---|---|---|
| CenterNet (Duan et al., 2019) |  |  |  |
| Faster RCNN (Girshick, 2015) |  |  |  |
| SSD Resnet (Liu et al., 2016) |  |  |  |
| YOLOv5s (Jocher et al., 2022) |  |  |  |
| YOLOv6 (Li, Li, et al., 2022) |  |  |  |
| YOLOv7 (Wang, Bochkovskiy & Liao, 2023) |  |  |  |
| YOLOv7-tiny (Cheng et al., 2023) |  |  |  |

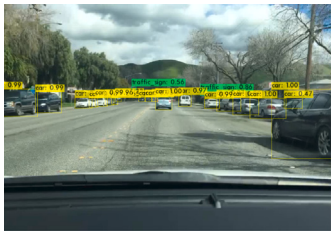
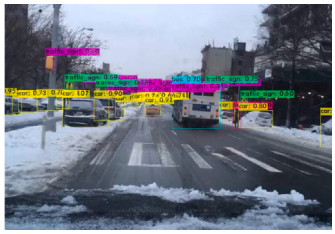
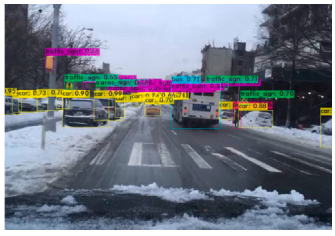
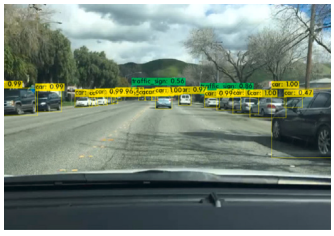
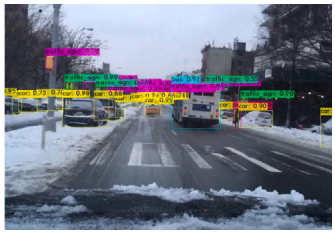
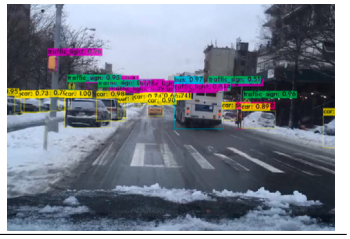
(continued on next page)

**Table 13** (*continued*).

| | | | |
|---|---|---|---|
| YOLOv8s (Vats & Anastasiu, 2023) |  |  |  |
| SD-YOLO-AWDNet |  |  |  |
| Models | Partly cloudy | Rainy | Snowy |
| CenterNet (Duan et al., 2019) |  |  |  |
| Faster RCNN (Girshick, 2015) |  |  |  |
| SSD Resnet (Liu et al., 2016) |  |  |  |
| YOLOv5s (Wang et al., 2022) |  |  |  |
| YOLOv6 (Li, Li, et al., 2022) |  |  |  |

(*continued on next page*)

**Table 13** (*continued*).

| | | | |
|---|---|---|---|
| YOLOv7 (Wang, Bochkovskiy & Liao, 2023) |  |  |  |
| YOLOv7-tiny (Cheng et al., 2023) |  |  |  |
| YOLOv8s (Vats & Anastasiu, 2023) |  |  |  |
| SD-YOLO-AWDNet |  |  |  |

### 4.4. Performance validation on two other datasets

The paper involves a major analysis of the BDD100k dataset as shown in previous sections. To validate the robustness of the model across diverse datasets, analysis is also done on other popular datasets including Canadian Adverse Driving Condition (CADC) (Pitropov et al., 2021) and Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) (Geiger et al., 2013). The following sections showcase comparison analysis with KITTI and CADC respectively.

#### 4.4.1. mAP comparision on CADC dataset

The comparison Table 11 shows the mAP@0.50, mAP@0.75, FPS, Giga Floating Point Operations Per Second (GFLOPs) and several parameters values of different object detection algorithms for the CADC dataset (Pitropov et al., 2021). SSD Resnet performed least with mAP of 0.12 with FasterRCNN and CenterNet also performing weaker with mAP values of 22.60 and 23.80. Variants of YOLOv5s, YOLOv6, YOLOv7 and YOLOv7-tiny performed slightly better with mAP values of 25.60, 25.90, 27.60 and 27.20, but still lagged behind the proposed model. YOLOv8s performed similarly to the proposed model with a mAP of 28.20, but the total parameters are significantly lesser in the proposed model. Together, these aspects highlight the lightweight design and increased performance efficiency of the proposed model SD-YOLO-AWDNet.

#### 4.4.2. mAP comparison on KITTI dataset

The comparison Table 12 shows the mAP@0.50, mAP@0.75, FPS, Giga Floating Point Operations Per Second (GFLOPs) and several parameters values of different object detection algorithms for the KITTI dataset (Geiger et al., 2013). CenterNet performed the least with a mAP of 29.2. Faster-RCNN improves the detection performance with a higher mAP value of 54.5 as compared to SSDResNet and CenterNet which performed relatively poorly with mAP values of 44.0 and 29.2. YOLOv5 improves detections with mAP of 83.3 but it is still lower than YOLOv7 and YOLOv8s with mAP of 84.91 and 84.99. The proposed model SD-YOLO-AWDNet with mAP of 86.0 outperforms the baseline model significantly with overall benchmarking across all key indicators, with the least number of training parameters.

An improved lightweight object detection model is developed for adverse weather conditions in self-driving cars, delivering improved FPS on existing hardware. Its adaptability to weaker hardware offers superior performance and efficiency compared to other models. These findings provide insights for updates to driving systems, facilitating multi-scale target detection in adverse weather and accelerating model deployment for low-end vehicles, thereby enhancing the feasibility of autonomous vehicles (ADS) for wider public distribution.

#### 4.4.3. SD-YOLO-AWDNet detections on out-of-distribution datasets

For training and validation purposes BDD100k, CADC and KITTI datasets are used. However visual analysis is done on real-world images from Indian road traffic. The collected images span across different places and weather scenarios i.e rainy, snowy, daytime, night-time, sunny, sparse traffic, dense traffic, valley and hill areas. This is done to validate the effectiveness of the proposed model on out-of-context datasets. It is evident from Fig. 17 that objects are detected more precisely by the proposed model on images of real world scenarios. In future, detailed analysis will be done for different traffic, various terrains and adverse weather conditions.

## 5. Conclusion

This paper presents SD-YOLO-AWDNet, improved lightweight YOLOv5s for enhanced detection in adverse weather conditions while reducing model size without sacrificing efficiency. It utilizes ghost convolutions with DSDC and CA modules, with a novel loss function for improved performance. Comparative analysis reveals SD-YOLO-AWDNet's performance gains for mAP 50 is [92%, 72.3%, 41.05%, 2.24%, 5.7%, 4.04%, 15.7%, and 1.77%] for BDD100k dataset, [57.7%, 97.4%, 78%, 3.24%, 3.12%, 1.28%, 2.22%, and 1.18%] for KITTI dataset, and [27.4%, 40%, 21%, 12.5%, 11.16%, 4.31%, 5.8%, and 2.12%] for CADC dataset in comparison to [Faster RCNN, SSD-ResNet, CenterNet, YOLOv5, YOLOv6, YOLOv7, YOLOv7-tiny, and YOLOv8] models. The performance shows that the proposed model successfully reduces [91.27%, 85.5%, 82.6%, 94.2%, 50.6%, 81.1%, 90%, 40.3% and 66.9%] number of parameters in comparison to [FasterRCNN, SSD ResNet, CenterNet, YOLOv4, YOLOv5, YOLOv6, YOLOv7, YOLOv7-tiny and YOLOv8s], respectively.

The future goal is to utilize this method with different object detection techniques to enhance model detection performance, alongside conducting deeper analyses on the CADC and KITTI datasets using the proposed model. A custom dataset will also be generated in future for further evaluating the model's efficiency.

## CRediT authorship contribution statement

**Rashmi:** Software, Formal analysis, Resources, Data curation, Writing – original draft, Visualization, Investigation, Software, Validation, Writing – review & editing. **Rashmi Chaudhry:** Conceptualization, Methodology, Supervision, Project administration, Funding acquisition, Writing – review & editing, Result analysis.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

The authors express their gratitude to the Center of Excellence in Artificial Intelligence Lab at Netaji Subhas University of Technology for providing the necessary resources and enabling the successful completion of the implementation work.

## References

Adarsh, P., Rathi, P., & Kumar, M. (2020). YOLO v3-tiny: Object detection and recognition using one stage improved model. In *2020 6th international conference on advanced computing and communication systems* (pp. 687–694). IEEE.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

Cai, Y., Luan, T., Gao, H., Wang, H., Chen, L., Li, Y., et al. (2021). YOLOv4-5D: An effective and efficient object detector for autonomous driving. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–13.

Chen, L., Ding, Q., Zou, Q., Chen, Z., & Li, L. (2020). DenseLightNet: A light-weight vehicle detection network for autonomous driving. *IEEE Transactions on Industrial Electronics*, *67*, 10600–10609.

Cheng, P., Tang, X., Liang, W., Li, Y., Cong, W., & Zang, C. (2023). Tiny-YOLOv7: Tiny object detection model for drone imagery. In *International conference on image and graphics* (pp. 53–65). Springer.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251–1258).

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6569–6578).

Gao, P., Ji, C.-L., Yu, T., & Yuan, R.-Y. (2024). YOLO-TLA: An efficient and lightweight small object detection model based on YOLOv5. arXiv preprint arXiv:2402.14309.

Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430.

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, *32*, 1231–1237.

Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 3354–3361). IEEE.

Girshick, R. (2015). Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440–1448).

Hamzenejadi, M. H., & Mohseni, H. (2023). Fine-tuned YOLOv5 for real-time vehicle detection in UAV imagery: Architectural improvements and performance boost. *Expert Systems with Applications*, Article 120845.

Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., & Xu, C. (2020). GhostNet: More features from cheap operations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1580–1589).

Hnewa, M., & Radha, H. (2020). Object detection under rainy conditions for autonomous vehicles: A review of state-of-the-art and emerging techniques. *IEEE Signal Processing Magazine*, *38*, 53–67.

Hnewa, M., & Radha, H. (2021). Multiscale domain adaptive yolo for cross-domain object detection. In *2021 IEEE international conference on image processing* (pp. 3323–3327). IEEE.

Hoque, S., Xu, S., Maiti, A., Wei, Y., & Arafat, M. Y. (2023). Deep learning for 6D pose estimation of objects — A case study for autonomous driving. *Expert Systems with Applications*, *223*, Article 119838.

Hou, Q., Zhou, D., & Feng, J. (2021). Coordinate attention for efficient mobile network design. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13713–13722).

Jiang, Y., Zhu, B., Zhao, X., & Deng, W. (2023). Pixel-wise content attention learning for single-image deraining of autonomous vehicles. *Expert Systems with Applications*, *224*, Article 119990.

Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., Kwon, Y., Fang, J., et al. (2022). ultralytics/YOLOv5: V6. 1-TensorRT, TensorFlow edge TPU and OpenVINO export and inference. *Zenodo*.

Johari, A., & Swami, P. D. (2020). Comparison of autonomy and study of deep learning tools for object detection in autonomous self driving vehicles. In *2nd international conference on data, engineering and applications* (pp. 1–6). IEEE.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., et al. (2022). YOLOv6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976.

Li, W., Liu, X., & Yuan, Y. (2022). Sigma: Semantic-complete graph matching for domain adaptive object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5291–5300).

Li, Z., Peng, C., Yu, G., Zhang, X., Deng, Y., & Sun, J. (2017). Light-head R-CNN: In defense of two-stage object detector. arXiv preprint arXiv:1711.07264.

Li, J., Xu, R., Ma, J., Zou, Q., Ma, J., & Yu, H. (2023). Domain adaptive object detection for autonomous driving under foggy weather. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 612–622).

Li, H., Zhuang, X., Bao, S., Chen, J., & Yang, C. (2024). SCD-YOLO: A lightweight vehicle target detection method based on improved YOLOv5n. *Journal of Electronic Imaging*, *33*, Article 023041.

Liao, W.-H., Wang, C.-C., & Lin, W.-C. (2023). GNN-based point cloud maps feature extraction and residual feature fusion for 3D object detection. In *2023 IEEE international conference on robotics and automation* (pp. 7010–7016). IEEE.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. (2016). SSD: Single shot multibox detector. In *Computer vision–ECCV 2016: 14th European conference, amsterdam, the netherlands, October 11–14, 2016, proceedings, part I 14* (pp. 21–37). Springer.

Liu, Z., Cai, Y., Wang, H., Chen, L., Gao, H., Jia, Y., et al. (2021). Robust target recognition and tracking of self-driving cars with radar and camera information fusion under severe weather conditions. *IEEE Transactions on Intelligent Transportation Systems*, *23*, 6640–6653.

Liu, G., Hu, Y., Chen, Z., Guo, J., & Ni, P. (2023). Lightweight object detection algorithm for robots with improved YOLOv5. *Engineering Applications of Artificial Intelligence*, *123*, Article 106217.

Liu, X., Ma, Y., Shi, Z., & Chen, J. (2019). GridDehazeNet: Attention-based multi-scale network for image dehazing. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 7314–7323).

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).

Liu, W., Ren, G., Yu, R., Guo, S., Zhu, J., & Zhang, L. (2022). Image-adaptive YOLO for object detection in adverse weather conditions. In *Proceedings of the AAAI conference on artificial intelligence*: *vol. 36*, (pp. 1792–1800).

Mahaur, B., Mishra, K., & Kumar, A. (2023). An improved lightweight small object detection framework applied to real-time autonomous driving. *Expert Systems with Applications*, *234*, Article 121036.

Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A. S., et al. (2019). Benchmarking robustness in object detection: Autonomous driving when winter is coming. arXiv preprint arXiv:1907.07484.

Ouyang, W., Wang, X., Zeng, X., Qiu, S., Luo, P., Tian, Y., et al. (2015). DeepID-Net: Deformable deep convolutional neural networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2403–2412).

Paz, D., Zhang, H., Li, Q., Xiang, H., & Christensen, H. I. (2020). Probabilistic semantic mapping for urban autonomous driving applications. In *2020 IEEE/RSJ international conference on intelligent robots and systems* (pp. 2059–2064). IEEE.

Pitropov, M., Garcia, D. E., Rebello, J., Smart, M., Wang, C., Czarnecki, K., et al. (2021). Canadian adverse driving conditions dataset. *The International Journal of Robotics Research*, *40*, 681–690.

Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for activation functions. arXiv preprint arXiv:1710.05941.

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).

Rezaei, M., Azarmi, M., & Mir, F. M. P. (2023). 3D-Net: Monocular 3D object recognition for traffic monitoring. *Expert Systems with Applications*, *227*, Article 120253.

Sakaridis, C., Dai, D., & Van Gool, L. (2018). Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, *126*, 973–992.

Shi, W., & Rajkumar, R. (2020). Point-GNN: Graph neural network for 3D object detection in a point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1711–1719).

Simhambhatla, R., Okiah, K., Kuchkula, S., & Slater, R. (2019). Self-driving cars: Evaluation of deep learning techniques for object detection in different driving conditions. *SMU Data Science Review*, *2*, 23.

Sindagi, V. A., Oza, P., Yasarla, R., & Patel, V. M. (2020). Prior-based domain adaptive object detection for hazy and rainy conditions. In *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part XIV 16* (pp. 763–780). Springer.

Sun, W., Zhang, X., & He, X. (2020). Lightweight image classifier using dilated and depthwise separable convolutions. *Journal of Cloud Computing*, *9*, 55.

Tabata, A. N., Zimmer, A., dos Santos Coelho, L., & Mariani, V. C. (2023). Analyzing CARLA's performance for 2D object detection and monocular depth estimation based on deep learning approaches. *Expert Systems with Applications*, *227*, Article 120200.

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).

Tian, Y., Gelernter, J., Wang, X., Li, J., & Yu, Y. (2019). Traffic sign detection using a multi-scale recurrent attention network. *IEEE Transactions on Intelligent Transportation Systems*, *20*, 4466–4475.

Tian, D., Han, Y., & Wang, S. (2024). Object feedback and feature information retention for small object detection in intelligent transportation scenes. *Expert Systems with Applications*, *238*, Article 121811.

Tian, Z., Shen, C., Chen, H., & He, T. (2019). FCOS: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9627–9636).

Vats, A., & Anastasiu, D. C. (2023). Enhancing retail checkout through video inpainting, YOLOv8 detection, and deepsort tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 5529–5536).

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2021). Scaled-YOLOv4: Scaling cross stage partial network. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition* (pp. 13029–13038).

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2023). 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* (pp. 7464–7475).

Wang, Z., Li, Y., Liu, Y., & Meng, F. (2024). Improved object detection via large kernel attention. *Expert Systems with Applications*, *240*, Article 122507.

Wang, D., Wen, J., Wang, Y., Huang, X., & Pei, F. (2019). End-to-end self-driving using deep neural networks with multi-auxiliary tasks. *Automotive Innovation*, *2*, 127–136.

Wang, H., Xu, Y., He, Y., Cai, Y., Chen, L., Li, Y., et al. (2022). YOLOv5-Fog: A multiobjective visual detection algorithm for fog driving scenes based on improved YOLOv5. *IEEE Transactions on Instrumentation and Measurement*, *71*, 1–12.

Wang, H., Xu, Y., Wang, Z., Cai, Y., Chen, L., & Li, Y. (2023). Centernet-auto: A multi-object visual detection algorithm for autonomous driving scenes based on improved centernet. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Wei, Y., Yuan, Q., Shen, H., & Zhang, L. (2017). Boosting the accuracy of multispectral image pansharpening by learning a deep residual network. *IEEE Geoscience and Remote Sensing Letters*, *14*, 1795–1799.

Xuan, W., Jian-She, G., Bo-Jie, H., Zong-Shan, W., Hong-Wei, D., & Jie, W. (2022). A lightweight modified YOLOX network using coordinate attention mechanism for PCB surface defect detection. *IEEE Sensors Journal*, *22*, 20910–20920.

Yang, M., & Fan, X. (2024). YOLOv8-Lite: A lightweight object detection model for real-time autonomous driving systems. *IECE Transactions on Emerging Topics in Artificial Intelligence*, *1*, 1–16.

Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., et al. (2020). BDD100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2636–2645).

Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122.

Zhang, C., Li, Z., Liu, J., Peng, P., Ye, Q., Lu, S., et al. (2021). Self-guided adaptation: Progressive representation alignment for domain adaptive object detection. *IEEE Transactions on Multimedia*, *24*, 2246–2258.

Zhang, Y.-F., Ren, W., Zhang, Z., Jia, Z., Wang, L., & Tan, T. (2022). Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing*, *506*, 146–157.

Zhang, Z., & Xian, C. (2021). Consistent depth prediction under various illuminations using dilated cross attention. arXiv preprint arXiv:2112.08006.

Zhang, Y., Zhang, H., Huang, Q., Han, Y., & Zhao, M. (2024). DSP-YOLO: An anchor-free network with DsPAN for small object detection of multiscale defects. *Expert Systems with Applications*, *241*, Article 122669.