electronics

MDPI

Article

Enhancing the Safety of Autonomous Vehicles in Adverse Weather by Deep Learning-Based Object Detection

Biwei Zhang, Murat Simsek, Michel Kulhandjian and Burak Kantarci *

School of EECS, University of Ottawa, Ottawa, ON K1N 6N5, Canada; bzhan138@uottawa.ca (B.Z.); murat.simsek@uottawa.ca (M.S.); mkulhand@uottawa.ca (M.K.)
* Correspondence: burak.kantarci@uottawa.ca; Tel.: +1-613-562-5800 (ext. 6955)

Abstract: Recognizing and categorizing items in weather-adverse environments poses significant challenges for autonomous vehicles. To improve the robustness of object-detection systems, this paper introduces an innovative approach for detecting objects at different levels by leveraging sensors and deep learning-based solutions within a traffic circle. The suggested approach improves the effectiveness of single-stage object detectors, aiming to advance the performance in perceiving autonomous racing environments and minimizing instances of false detection and low recognition rates. The improved framework is based on the one-stage object-detection model, incorporating multiple lightweight backbones. Additionally, attention mechanisms are integrated to refine the object-detection process further. Our proposed model demonstrates superior performance compared to the state-of-the-art method on the DAWN dataset, achieving a mean average precision (mAP) of 99.1%, surpassing the previous result of 84.7%.

Keywords: object detection; adverse weather; self-driving cars; YOLOv5; neural network architecture; single shot detection

Citation: Zhang, B.; Simsek, M.; Kulhandjian, M.; Kantarci, B. Enhancing the Safety of Autonomous Vehicles in Adverse Weather by Deep Learning-Based Object Detection. *Electronics* **2024**, *13*, 1765. https://doi.org/10.3390/electronics13091765

Academic Editor: Piotr Borkowski

Received: 21 March 2024
Revised: 18 April 2024
Accepted: 23 April 2024
Published: 2 May 2024

1. Introduction

Recently, the utilization of sensory data from connected vehicles to capture contextual information has been advanced through the long short-term memory (LSTM)-based auto-encoder network [1,2]. As sensors play a crucial role in detecting and interpreting environmental data in robots and self-driving vehicles, a multi-sensory and multilevel enhanced convolutional network structure model is presented in the work of [3]. The enhancement strategy involves refining the network architecture to optimize feature fusion for drone object-detection algorithms, as detailed in [4]. Furthermore, the ML-YOLOv5 [5] is introduced for insulator defect detection. This approach is built upon the you only look once (YOLO), specifically version 5, network architecture. As illustrated in Figure 1, this paper introduces a novel framework for multi-sensory object detection in road scenes. We improve upon the single-stage detector, YOLO. YOLO, an anchor-less architecture, has achieved a breakthrough in object detection by treating the problem as a simple regression task. Utilizing a one-stage detector in a weather-adverse dataset proves advantageous for addressing object model challenges in terms of speed, network comprehension of generalized object representation, and a faster approach. In summary, our primary contributions can be outlined as follows:

- Propose a one-stage object-detection module, YOLOv5, incorporating multiple lightweight backbones such as ShuffleNetV2 [6], GhostNet, VoVNet [7], and computer vision attention mechanisms, including squeeze-and-excitation (SE) block [8], convolution block attention module (CBAM) block [9], or efficient channel attention (ECA) block [10] to advance the performance in detecting objects in challenged settings.
- Integrate transformer prediction heads (TPH) into YOLOv5 to enhance object localization, particularly for adverse scenes.

- Simplify the framework with the Pruning method, quantization, and distillation specifically tailored for addressing adverse weather-related issues to reduce the computational cost and size of the model.
- Augment YOLOv5 with CBAM to improve the network's capacity to recognize regions of interest in photos with broad region coverage. For enhanced classification of object categories, a self-trained classifier is employed. Our proposed framework achieves a remarkable 99.7% average precision (AP), surpassing the baseline method of [11] by 5%.

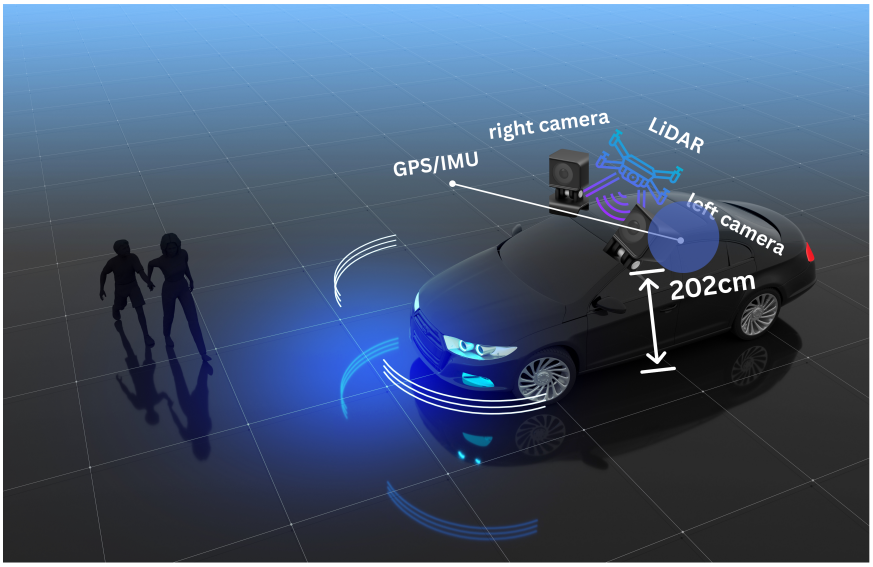

Figure 1. The sensor configuration for collecting A2*D and 3D vehicle platform data on the A*STAR autonomous driving vehicle includes a spinning Velodyne LiDAR and two color PointGrey Chameleon3 cameras positioned on either side of the LiDAR.

Consequently, the suggested framework appears as a promising solution for enhancing a wide range of vision-based applications, even in adverse weather conditions. It is worth noting that its potential spans diverse domains such as UAV-based object detection [12,13], pedestrian safety alert systems [14,15], and intelligent transportation systems reliant on vehicle-to-infrastructure (V2I) communication [16]. Our approach is capable of overcoming the challenges posed by adverse weather, thereby introducing enhanced performance and reliability across these critical applications.

2. Related Work

This paper commences by reviewing and discussing conventional object-detection algorithms documented in the literature. Subsequently, it delves into a discourse on object-detection algorithms specifically tailored for scenarios characterized by poor visibility. Finally, the timeline of the presented state-of-the-art solutions in this study for object detection are provided in Figure 2.

2.1. Traditional Object-Detection Algorithms

Object detection typically employs two methods: one-stage and two-stage detection methods. In the single-stage approach, the technique directly predicts bounding boxes and the probability of classes for the targets. On the other hand, the two-stage approach involves the algorithm initially generating a set of region proposals and subsequently classifying those proposals as either objects or backgrounds.

The two-stage detection model, often referred to as the multiple-stage detection model, is one of the most studied aspects in the field of object detection. Notably, fast region-based convolutional neural network (Fast R-CNN) and Mask-R-CNN belong to the widely used

family of two-stage object detectors within the R-CNN architecture. Despite the R-CNN series generally yielding good results in object-detection accuracy, the two-stage detector presents challenges such as prolonged training times, increased inference durations, and higher computational costs.

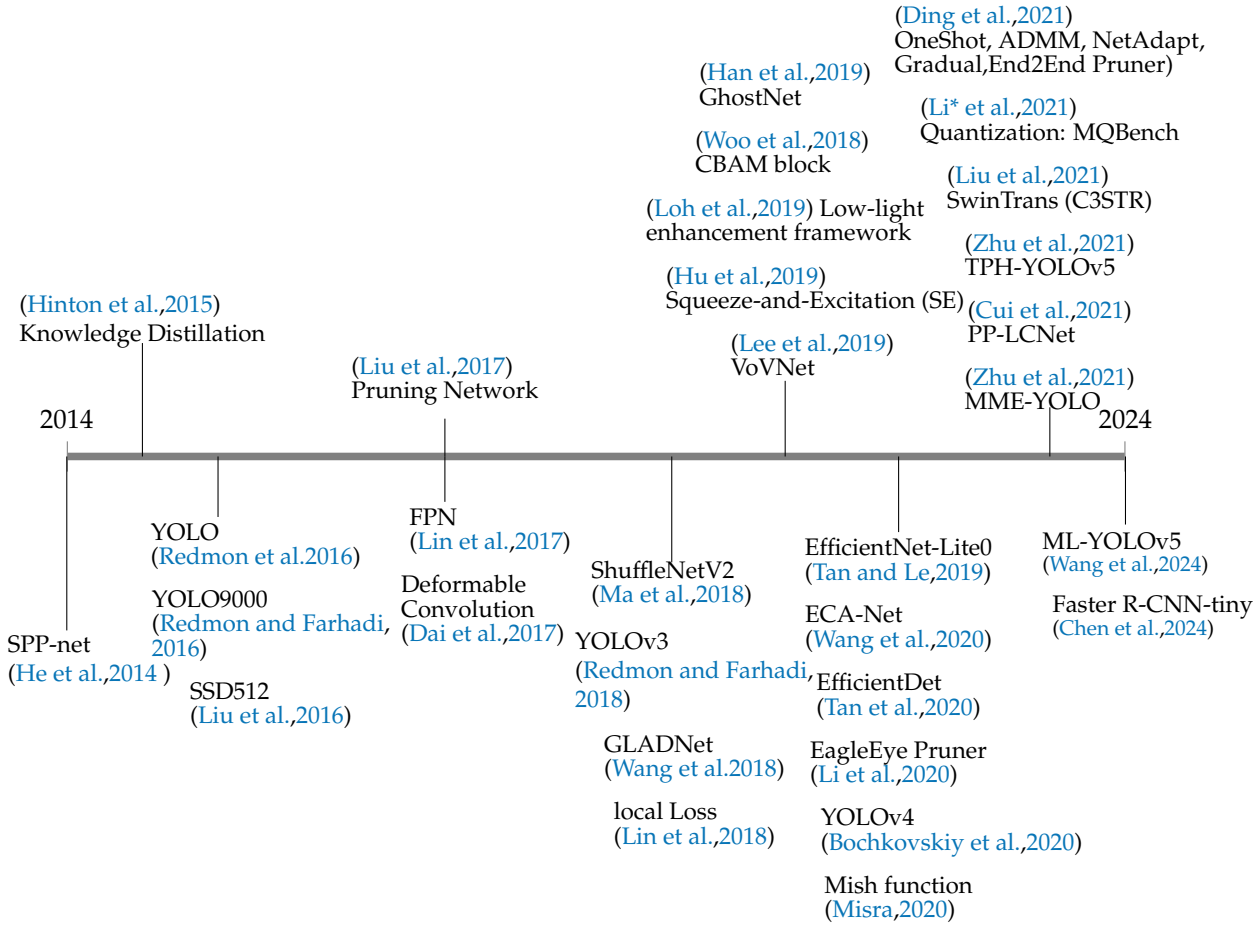

Figure 2. Brief history of presented algorithms with the timeline ([3–10,17–38]).

Single-stage object-detection methods such as YOLO, single-shot multiBox detector (SSD) [17], EfficientDet [18], and RetinaNet [19] typically employ a single fully CNN (FCNN) to simultaneously detect objects' classes and spatial locations without intermediate steps. This stands in contrast to two-stage object-detection methods like Fast R-CNN.

Among various single-stage object-detection methods, YOLOv5 [39] has garnered significant interest since its introduction in 2016. The fundamental concept of YOLO is to partition an input image into a matrix of individual cells and predict bounding boxes and class probabilities for each cell. YOLOv1 [20] featured a simple structure with two fully interconnected layers at the back and twenty-four convolutional layers for delivering probabilities and coordinates. Since its inception, YOLO has undergone several improvements and iterations. In 2017, YOLOv2 [21] was introduced with performance enhancements achieved through multi-scale training, anchor boxes, batch normalization, Darknet-19 architecture, and a modified loss function. Subsequently, Redmon et al. introduced YOLOv3 [22], which incorporated a feature pyramid network, convolutional layers with anchor boxes, spatial pyramid pooling (SPP) block [23], Darknet-53 architecture, and an improved loss function. In contrast to previous versions, YOLOv4 was introduced by different authors, A. Bochkovskiy et al. [24], enhancing YOLO's performance through the utilization of CSPDarknet53 architecture, Bag-of-Freebies, Bag-of-Specials, Mish activation function [25], and weighted-residual-connections [40].

2.2. Object-Detection Algorithms in Low Visibility

Employing consistent enhancement techniques and fine-tuning enhancement functions proves advantageous for scene recognition in limited visibility conditions. Image enhancement techniques fall into two categories: those reliant on pixel manipulation and those leveraging features. These techniques often involve the transformation of images using CNNs. However, applying uniform enhancement techniques across an entire image may not consistently yield optimal results, given the non-uniformity of luminance present in a scene. Scenarios with multiple light sources may necessitate adjusting enhancement functions to achieve effective feature retrieval, adding complexity to the process. Current research efforts addressing this challenge include GLADNet [26], a low-light enhancement network incorporating global awareness. Additionally, ref. [27] introduces Gaussian processes regression (GPR) to create a distribution of localized feature-enhancement functions, with CNNs providing support for the process.

In contrast, object detectors based on deep learning exhibit remarkable effectiveness in tasks related to object detection, even under challenging low-light conditions. Recent research endeavors aimed at addressing this issue include RetinaNet [19], a deep learning model specifically crafted for object identification problems. RetinaNet utilizes a feature pyramid network (FPN) [28] to extract features from images across various scales. Distinguished by its innovative focal loss function, RetinaNet effectively tackles the class imbalance challenge inherent in object detection. This model demonstrates notable accuracy in object-detection tasks, even in low-light conditions.

Additionally, designed explicitly for object recognition, the SSD presented in [17] is a deep neural network introduced by Liu et al. This model predicts rectangular region annotations and class probabilities of objects in an image using a single neural network. SSD strategically employs multi-scale feature extraction to identify objects across diverse scales and sizes. Moreover, the application of pruning and quantization techniques can prove beneficial in optimizing object-detection performance, particularly in adverse weather conditions.

3. Methodology

3.1. Overview of Proposed One-Stage Approach

This paper aims to enhance the one-stage detection methodology of YOLOv5 by modifying the structure of the models across all scales. This section presents all the work related to pruning and quantification in YOLOv5, accompanied by the results obtained from various perspectives.

In the initial training session, YOLOv5 was selected as the baseline model due to its advantages in terms of speed, real-time performance, and the efficiency of its single forward pass methodology.

In YOLOv5, the image is partitioned into a grid, and the grid cell containing a specific object is identified. This designated grid cell is responsible for detecting that particular object, involving the prediction of bounding boxes based on the confidence scores of each grid cell. Additionally, the model predicts the probability of conditional classes. Refer to Figure 3 for an illustration of the steps involved in how YOLOv5 operates.

In implementing baseline models, we have opted not to apply data augmentation techniques during the data preparation stage. This decision is made to avoid introducing biases from the existing dataset into the augmented dataset. Various data augmentation operations can alter the data distribution throughout the training process. Despite numerous positive improvements attributed to data augmentation, it is not guaranteed to necessarily enhance generalization errors [41]. Especially, research in [42] suggests that training with augmented data may yield only a modest improvement in robust error while potentially resulting in a significant increase in standard error.

As depicted in Figure 4, the workflow of the proposed methodology is presented to facilitate a clear understanding of the content discussed from Sections 3.4–3.7. A detailed view of the structural framework of the proposed one-stage detection module is provided in Figure 5. Our investigation encompasses an in-depth examination of the fundamental

structure, incorporating discussions on heads, necks, and backbones. Various approaches are explored for model integration, aiming to enhance performance through the utilization of state-of-the-art detectors.

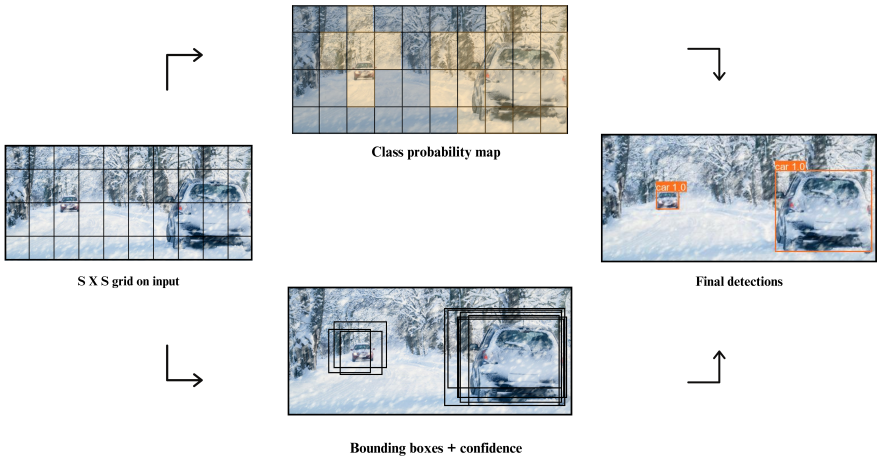

Figure 3. The procedure outlines the workflow of YOLOv5. Each image is segmented into uniformly sized boxes, and bounding boxes are drawn. The width of each line within the box corresponds to the confidence ratings.

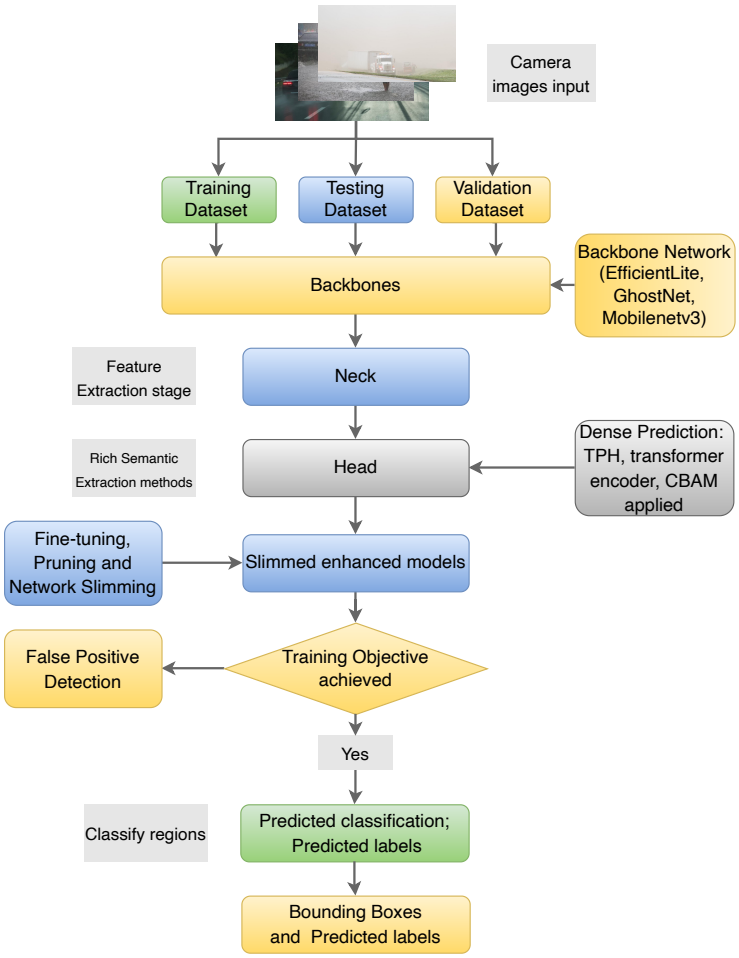

Figure 4. Workflow in the proposed one-stage detection model framework.

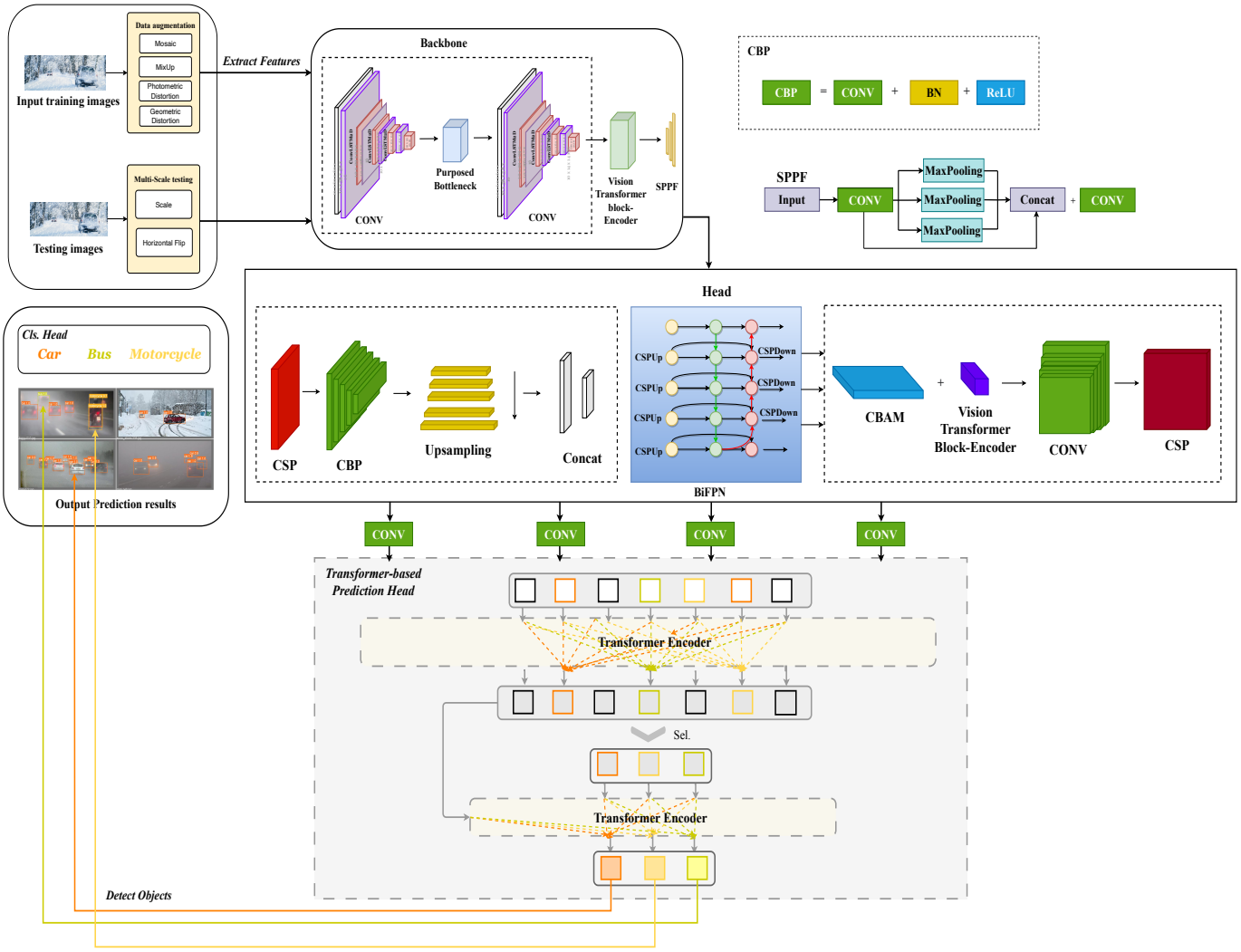

Figure 5. Structural framework of the proposed one-stage detection module. The framework introduces several novel features: (1) the removal of the CBAM module at positions 14 and 19 in Figure 6; (2) a reduction in the number of channels associated with the P2 head in the original YOLOv5, from 256 to 128; (3) the addition of an SPP layer between the backbone and the neck; (4) the execution of output after the CBAM module; (5) the retention of only the backbone and the last layer of the output TransBlock; and (6) the adoption of BiFPN as the neck. These alterations are aimed at achieving a lighter implementation, effectively reducing the parameters of the convolutional layer.

3.2. Component of the Enhanced Model

This work introduces several enhancements to the YOLOv5 structures aimed at strengthening the model's perceptual abilities and improving detection accuracy in practical scenarios. The characteristics detailing all the elements utilized in the implementation of the proposed model are presented in Table 1. This includes the Backbone Network, Detection Head, Training data, Preprocessing, Loss Function, technical specifications, implementation details, and other relevant characteristics of the model. Firstly, advanced prediction heads are explored under the original YOLOv5 to identify objects at various scales. The Transformer prediction head [43] is integrated to replace the original prediction heads, leveraging the self-attention mechanism's predictive capacity [44]. Additionally, the CBAM [9] is incorporated to detect focus regions within cluttered environments. Figure 6 illustrates how the new framework integrates a TPH [28] into the CBAM [9]. The term "four heads for predictions" refers to the small, medium, and large heads.

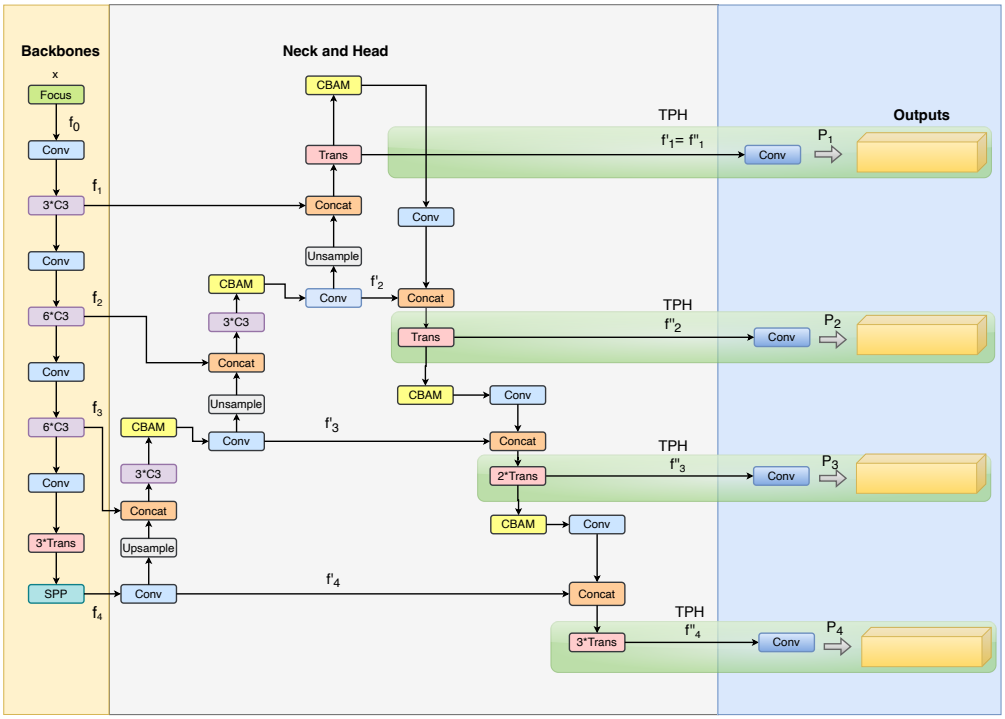

Figure 6. The architecture of the improved YOLOv5 is showcased with blue highlights indicating the integrated blocks.

Table 1. Characteristics of the method.

Model Architecture	**YOLOv5 (You Only Look Once)**
Backbone Network	The feature extraction network used as the backbone of the object-detection model, like GhostNet, EfficientNet, MobileNet, etc.
Detection Head	Detection layer 1: $80 \times 80 \times 256$ for detect small size object. Detection layer 2: $40 \times 40 \times 512$ for detect medium size object. Detection layer 3: $20 \times 20 \times 512$ for detect large size object.
Training Data	The DAWN dataset.
Preprocessing	The novel methodology that addresses these issues without altering the original images such as resizing, normalization, data augmentation, etc.
Loss Function	The function used to measure the difference between predicted and ground truth bounding boxes and class probabilities during training, like Binary Cross-Entropy with Logits Loss, Focal Loss function.
Optimization Algorithm	The algorithm used to optimize the model's parameters during training, such as stochastic gradient descent (SGD), Adam, RMSProp, etc.
Hyper-parameters	Tunable parameters that control the learning process, learning rate is default as 0.01, batch size is 2, number of epochs is 450/500, etc.
Evaluation Metrics	Metrics used to evaluate the performance of the model, such as mean Average Precision (mAP), Intersection over Union (IoU), precision, recall, etc.
Inference Method	We implemented multi-batch inference which optimizes deep learning models and accelerates inference in NVIDIA hardware and examines the difference in speed using the proposed model.

For analytical purposes, we establish the mathematical representation of the flow chart and model structure of YOLOv5. By denoting the input photos as x, we define f_i as $Focus(x)$. The output from the backbone, comprising four features, is expressed as

$$f_i = B_i(f_{i-1}), \quad i \in \{1, 2, 3, 4\}, \tag{1}$$

where B_i denotes various blocks in the backbone. Specifically, B_i for $i = 1, 2, 3$ signifies the combination of convolutional layers, involving 3, 6, or 9 CSPbottleneck modules. The composition of B_4 involves three transformer modules, an SPP module, and a convolutional layer [23]. In the neck section, four features are represented as f_i', which can be expressed as

$$f_i' = \begin{cases} N_i(f_i, f_{i+1}'), & i \in \{1, 2, 3\} \\ Conv(f_i), & i = 4 \end{cases}, \tag{2}$$

where $Conv()$ represents convolutional layers, and N_i represents different blocks expressed in the manner described below:

$$N_i(f_i, f_{i+1}') = UpBlock(Concat(f_i, Upsampling(f_i' + 1))). \tag{3}$$

The synthesis of several modules is represented by the UpBlock. In N_2 and N_3, the UpBlock comprises three cross-stage partial (CSP) bottleneck modules, a CBAM module, and a convolutional layer. Meanwhile, in N_1, the CBAM and transformer modules are included within the UpBlock [9]. The features listed prior to the last four convolution layers are derived as follows

$$f_i'' = \begin{cases} f_1', & i = 1 \\ H_i(f_i, f_{i-1}''), & i \in \{2, 3, 4\} \end{cases}, \tag{4}$$

where H_i denotes distinct blocks defined as

$$H_i(f_i, f_{i-1}'') = DownBlock(Concat(f_i, Conv(f_{i-1}''))), \text{for } i \in \{1, 2, 3, 4\}. \tag{5}$$

The DownBlock signifies the amalgamation of various modules. A CBAM module plus one, two, or three transformer modules are present in H_2, H_3, and H_4 of the DownBlock. Based on the acquisition of f_i'', the prediction can be obtained as follows

$$p_i = Conv(f_i''), \quad i \in \{1, 2, 3, 4\}, \tag{6}$$

where p_i represents the set of four output predictions derived from various prediction heads. In conclusion, we employ a methodology distinct from that of Convolutional Neural Networks (CNNs) in the component of the enhanced model. Bounding box and class probabilities are directly predicted from feature maps using convolutional layers rather than fully linked layers. A 3D tensor with the bounding box coordinates, objectness scores, and class predictions for every grid cell is produced using 1×1 convolutions at the network's conclusion.

3.3. Proposed Small Objects Prediction Head

As elucidated in Section 3.2, an additional prediction head, namely the transformer prediction head [35], is integrated into the YOLOv5 architecture to enhance the effectiveness and precision of small object detection in adverse weather scenarios.

To construct the prediction head of transformers, as depicted in Figure 6, a high-definition and lower-level feature map, more attuned to smaller objects, is utilized. Despite an increase in computation and memory costs, the incorporation of the detection head has resulted in improved performance in detecting small objects. In comparison with the original YOLO architecture's other three prediction heads, the four-head structure mitigates the negative impact of violent object scale variance. Consequently, the performance in detecting object classes such as bicycles and pedestrians, categorized as small objects, is notably enhanced by the novel transformer prediction head. The supplementary head augments sensitivity to small objects by amalgamating multi-layer superior characteristics with inferior high-quality data as input.

3.4. Transformer Encoder Block

Built upon the CSPDarknet53 design, YOLOv5 adopts an SPP as its structure and a PANet as its neck and head for YOLO detection. Drawing inspiration from [43], transformer encoder blocks are employed to replace YOLOv5's convolutional and CSP bottleneck components. The framework is illustrated in Figure 6.

The transformer encoder block serves to capture comprehensive global information and contextual details, a departure from the original bottleneck block in CSPDarknet53. Leveraging its attention mechanism, the block explores diverse possibilities for feature representation. Transformer encoder blocks consist of two key components: an MLP, or a feed-forward neural network, along with a multi-head attention block. LayerNorm and Dropout layers are incorporated to enhance convergence and mitigate overfitting in the network. The multi-head attention block aids each node in focusing on its pixels and comprehending the context, rendering transformer encoder blocks particularly effective in densely packed, enclosed environments. In alignment with Zhu et al.'s findings on the VisDrone2021 dataset in [45], the transformer encoder block in the CSPDarknet53 backbone, as opposed to the original bottleneck block, excels at capturing extensive contextual and global information.

3.5. Swin Transformer Block

In order to tackle the challenges posed by objects of diverse scales and achieve faster inference times per instance, we incorporate the Swim-Transformer into YOLOv5 architectures, presenting a novel network tailored for detection issues in adverse weather conditions. The Swin Transformer generates a hierarchical representation by progressively integrating neighboring small patches (outlined in black) into deep transformer layers, as depicted in Figure 7.

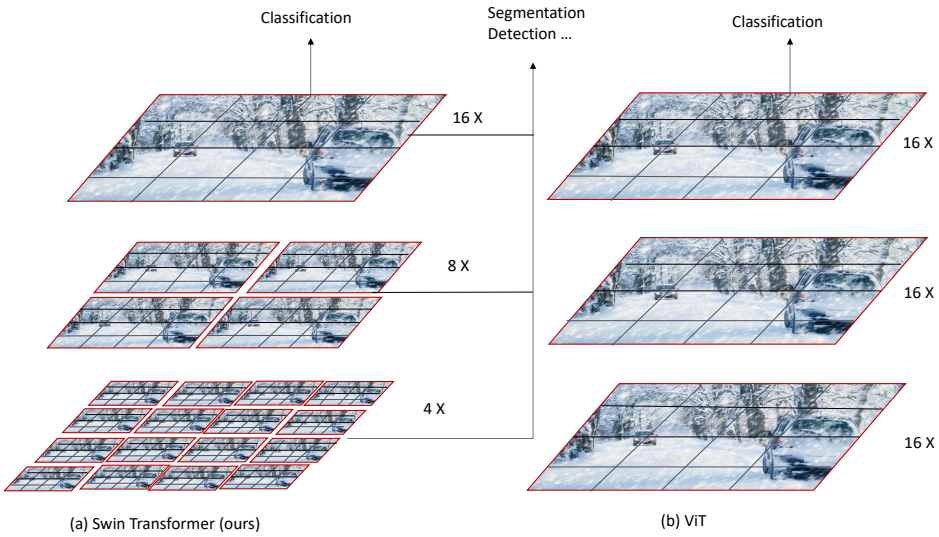

Figure 7. The architecture of the transformer encoder in the enhanced model. Hierarchical feature maps (depicted in black) are generated at deeper levels by the Swin Transformer through the integration of image patches. The approach confines self-attention computations within each local window, leading to linear computational complexity concerning the input image size (illustrated in red). This adaptable framework is well-suited for challenges involving dense recognition and image categorization.

The input RGB image undergoes segmentation using a patch partition with the Swin architecture into small non-overlapping patches [38]. Each patch is treated as a distinct entity, and its characteristics are defined by concatenating the raw pixel values of the RGB image. With a patch size of 4×4, 48 features (or $4 \times 4 \times 3$) are obtained for each patch.

The raw value feature, denoted as *C* in Figure 8, is projected to a random dimension using a linear embedding layer.

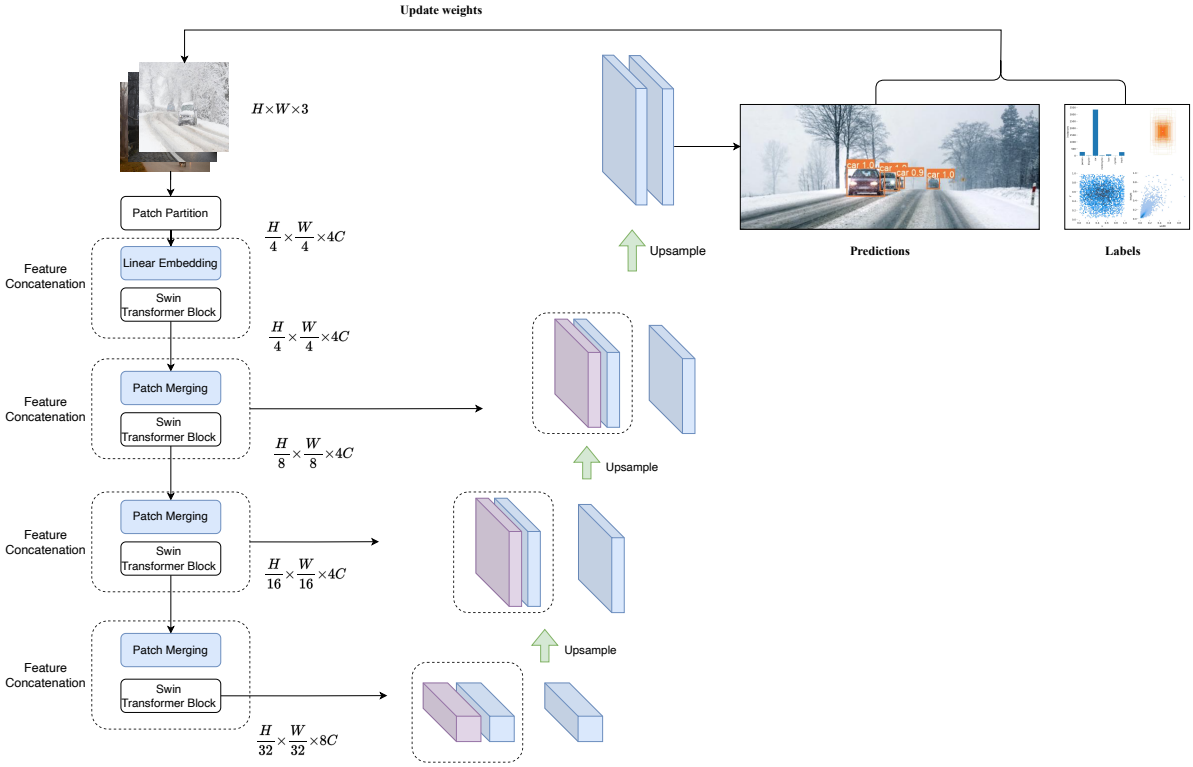

Figure 8. The diagram of the Swin Transformer in the proposed layout. In simulating the interactions between feature representations, multi-head self-attention is crucial for capturing the inter-feature linkages inside the Transformer.

The decoder initiates the process by employing upsampling blocks, starting with the lowest-resolution feature map. It upsamples the feature map, followed by concatenation with the corresponding skip connection from the encoder. The decoder comprises three blocks: a 3×3 convolutional layer, a ReLU layer, and a $2\times$ upsampling operator. The encoding of the decoder's features with the skip connection aligns with the U-Net design.

The cascaded upsampling method is employed to restore the prior layer's resolution. In the encoder phase, feature maps are generated, and multiple cascaded upsampling blocks are used to achieve the complete resolution. Each block has an $H \times W$ resolution with an upsampling layer, Batch normalization, ReLU, and two 3×3 convolution layers. The combined performance of the encoder and decoder creates a conventional U-shaped architecture, as depicted in Figure 8, enabling feature aggregation at various resolution levels through skip connections. Consequently, the encoder's final output consists of multi-level feature maps with higher-level features at greater levels but lower resolution than the previous one.

3.6. Convolutional Block Attention Module

The CBAM, a straightforward attention module designed to enhance the capability of deep neural networks, was initially introduced by Woo et al. in [9]. Consequently, the integration of CBAM into YOLOv5 enhances the network's ability to detect regions of interest in images with wide coverage. CBAM is a lightweight module that can be seamlessly trained and incorporated into popular CNN architectures. When presented with a feature map, CBAM constructs attention maps for two distinct dimensions—surface and channel—multiplying them by the input feature map to enact feature adjustments.

As outlined in [9], for a given feature map, CBAM sequentially derives attention maps based on two dimensions: spatial and channel. These maps are then multiplied by the input feature map. The CBAM module structure integrated into our framework is illustrated in Figure 9. The CBAM attention mechanism module within YOLOv5 demonstrates its effectiveness. Deploying CBAM facilitates the extraction of attention areas to support our detector and enhances object-detection accuracy, especially for small target objects.

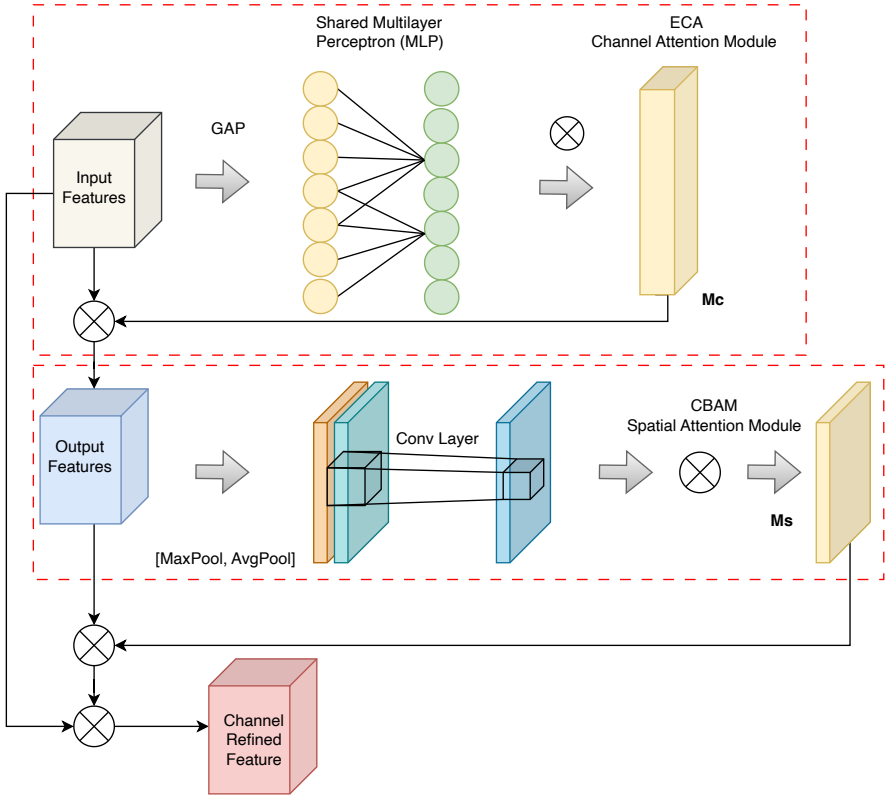

Figure 9. The architecture of CBAM in the enhanced YOLOv5 involves the utilization of two consecutive sub-modules, with the additional incorporation of residual paths.

3.7. Pruning Procedure and Strategy

The proposed methodology leverages EagleEye [33], a neural network pruning method, to optimize performance in low-visibility environments. EagleEye serves as an effective assessment tool that utilizes adaptive batch normalization to establish a robust relationship between various pruned deep neural network (DNN) architectures and their corresponding accuracy levels. To minimize the parameter count and computations of the YOLO model, we employ the EagleEye pruning method to enhance the performance of the proposed detector. The strong correlation provided by EagleEye allows us to efficiently prune candidates with the highest potential accuracy without the need for additional adjustments. In our experiments, EagleEye outperforms other pruning algorithms, as illustrated in Figure 10. This approach reveals the potential of sub-networks, enabling the selection of the most suitable candidates for pruning and implementation of the pruning strategy.

As discussed in [33], in ImageNet trials with an overall 50% reduction in operations (FLOPs), EagleEye achieves maximum accuracy of 70.9% using a compact representation of MobileNetV1 [46]. This result is 1.3% to 3.8% higher than accuracy achieved by other approaches.

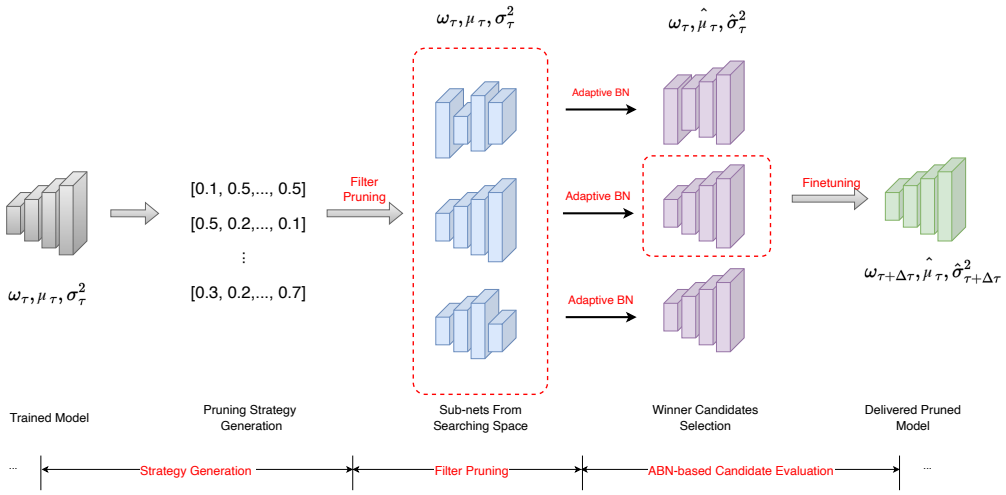

Figure 10. The pipeline of EagleEye Pruner.

3.8. Network Slimming

Drawing inspiration from network slimming [47], we present a pioneering-learning scheme for YOLOv5 that strives to simultaneously reduce the model size and lower the run-time memory footprint while maintaining accuracy.

The proposed method is tailored to the YOLOv5 architecture, minimizing training overhead and eliminating the need for specific software or hardware accelerators for each model, as compared to novel approaches. This technique identifies unimportant channels during the training of wide and massive networks, subsequently pruning them to generate thin, compact models with comparable accuracy.

Despite using the YOLOv5 baseline, our proposed YOLOv5, augmented with a slimming pruner and optimization solution, enhances object identification accuracy. The core concept involves scaling the output from each channel by the γ scaling factor [47]. Subsequently, we optimize scaling factors and network weights with a focus on sparsity regularization. This process prunes channels characterized by small factors, followed by adjusting the pruned network. The optimization problem is formulated as follows:

$$L = \sum_{(x,y)} l(f(x, W), y) + \lambda \sum_{\gamma \in \Gamma} g(\gamma), \tag{7}$$

where the first term $l()$ represents the general training loss of a convolutional neural network, $g()$ is a penalty term resulting from the sparsity of the scaling factor, and γ balances the two terms. The training input and target are denoted by x and y, respectively, whereas the trainable weights are denoted by W. In the experiment, we employ the ℓ_1 norm with $g(s)$ equal to the absolute value of s. This norm is frequently employed to achieve sparsity [48]. Sub-gradient descent is used as the optimization technique with the ℓ_1 penalty period. To avoid applying a sub-gradient at an irregular position, an alternative approach is to substitute the ℓ_1 penalty, with the smooth ℓ_1 penalty, as proposed by Yuan et al. [49]. Pruning efficiently eliminates all inbound and outbound connections to a channel, delivering superior performance in pruning compared to more complicated techniques.

4. Experimental Details

4.1. Setup

The training is conducted on a GPU machine equipped with 2 NVIDIA-SMI 470.63.01 GPUs, each with 62.5 GiB VRAM. Every experiment is completed on a machine with a 5-core Intel(R) Xeon(R) W-2295 CPU clocked at 3.00 GHz and an RTX 3070 graphics card. The experimental setup employs PyTorch 1.8.1, Python 3.8.8, and CUDA 11.4.

In this experiment, a batch size of 4 and an input picture size of 512 × 512 pixels are used, with a learning rate set at 0.01. The training epoch is set to 450/500 rounds,

utilizing Adam as the optimizer and the sigmoid activation unit as the activation function. The intersection over union (IoU) detection threshold for validation is set at 0.20. Empirical data is collected and recorded for subsequent statistical analysis.

4.2. Performance Measurement

Our proposed one-stage detection framework has been assessed using the baseline YOLOv5 results. Initially, the enhanced YOLOv5 is trained on our redefined data, and the outcomes are compared with those of a corresponding baseline model.

4.3. Dataset

The detection in adverse weather nature (DAWN) dataset [48] has been employed in this study. This dataset showcases a diverse range of traffic environments, including city, freeway, and highway scenarios, along with various traffic patterns.

The DAWN dataset consists of 1000 photographs captured in traffic environments, categorized into four meteorological conditions: fog, snow, rain, and sandstorm. The dataset was randomly split into training, testing, and validation sets with random splitting with Monte Carlo Cross-Validation [47] in Scikit-learn [50]. Random splitting is a suitable method to ensure all types of all four weather conditions: fog, snow, rain, and sandstorm are included in all training stages. Table 2 provides an overview of the redefined dataset used for the experiments.

Table 2. DAWN dataset: categories and quantities.

		Objects					
Training Stages	**Number of Images**	**Person**	**Bicycle**	**Car**	**Motorcycle**	**Bus**	**Truck**
Training set	328	162	550	1515	0	21	91
Validating set	106	28	193	226	9	5	14
Testing set	71	9	74	203	1	0	7
Total	505	199	817	1944	10	26	112

4.4. Evaluation Metrics

Following the training process, we will conduct a comparative analysis of precision, recall, F1 Score, confusion matrix, IoU, and precision-recall (PR) curve to assess the performance. Our proposed framework relies on YOLOv5, primarily employed for multi-class object detection as opposed to binary classification. YOLO directly forecasts bounding boxes and class probabilities for multiple objects within an image. Consequently, ROC or AUC curves are infrequently employed for assessing YOLO or analogous object-detection models. Instead, evaluating object-detection tasks typically utilizes metrics such as Mean Average Precision (mAP), Intersection over Union (IoU), Precision, Recall, and F1-score. These metrics determine the model's proficiency in precisely localizing objects and accurately classifying them across various classes.

5. Results and Discussion

The training process has been divided into two sessions. The initial session involves training with YOLOv5 models and the enhanced methods. YOLOv5 has four models, described as small (s), medium (m), large (l), and extra-large (x) models.

Considering the set of modifications integrated into the enhanced module, each alteration may yield varied effects depending on the dataset utilized. Therefore, the focus of our research is to demonstrate the outstanding performance results achieved by our framework. Our objective is to exceed the benchmarks established by the study conducted in [11]. To compare the results of YOLOv5 baseline with the enhanced methods, each model offers distinct levels of detection accuracy and performance as depicted in Tables 3 and 4, respectively. The preliminary findings from the original YOLOv5 [39] are presented first, with the confusion matrix for YOLOv5 displayed in Figure 11.

Table 3. The performance results of YOLOv5 baseline training process.

Methodology	Precision (%)	Recall (%)	F1	$mAP_{0.5}$ (%)	$mAP_{0.95}$ (%)
YOLOv5—L	0.85	0.90	0.86	0.90	0.90
YOLOv5—M	0.92	0.81	0.88	0.93	0.74
YOLOv5—N	0.87	0.80	0.91	0.90	0.74
YOLOv5—S	0.89	0.83	0.80	0.88	0.75
YOLOv5—X	0.90	0.85	0.80	0.90	0.69

Table 4. The performance results during the enhanced YOLOv5 training process *.

Methodology	Precision (%)	Recall (%)	F1	$mAP_{0.5}$ (%)	$mAP_{0.95}$ (%)
TPH-Slimming Pruned (Our)	0.94	0.81	0.89	96.8	76.1
YOLOv5x (Our)	0.75	**0.99**	0.66	**99.1**	**90.0**
YOLOv5s (Our)	0.95	0.81	0.88	98.5	80.9
YOLOv5s+YOLOv5xP2CBAM (Our)	**0.97**	0.80	**0.91**	98.1	73.5
YOLOv5xP2+YOLOv5s (Our)	0.90	0.83	0.80	97.2	75.1
YOLOv4-tiny [11]	0.82	0.52	0.64	49.9	29.1
YOLOv3-tiny [11]	0.80	0.59	0.68	51.3	25.4
YOLOv5-baseline [11]	0.91	0.82	0.90	84.7	65.1

* The bold characters represent all the best results of each evaluation criterion.

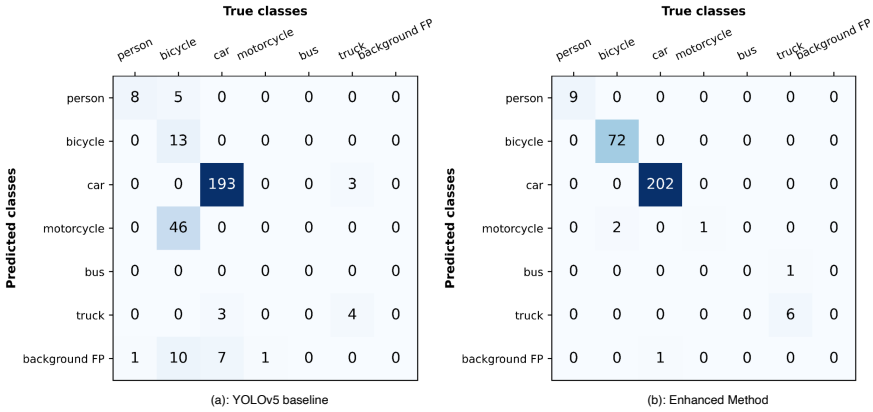

Figure 11. The confusion matrix comparison for baseline and enhanced methods.

5.1. Model Training

The results of the performance of the proposed method demonstrate real-time outcomes, gathered after the training phase. By initializing the parameters with COCO object recognition train models, we aim to compare the outcomes of the baseline model with the suggested approach. Subsequently, we will perform further fine-tuning of all parameters using our training data.

5.2. Performance Results

In Figure 12, the highest F1 value of 0.90 is achieved at a confidence value of 0.666, optimizing accuracy and recall. The graph illustrates an increase in confidence values and F1 scores as the epoch reaches 500, with a continuous improvement in the mean average precision (mAP)@0.5:0.95 index. In this experiment, we initially trained the input data with the YOLOv5 using default parameter settings. Subsequently, we trained the enhanced YOLOv5 algorithm with various backbones and necks. Remarkably, the enhanced method, tailored for smaller target sizes, demonstrated superior precision values and mAP compared to alternative configurations. The results of the enhanced method outperformed other advanced object-detection systems, as indicated by the comparison and analysis graphs.

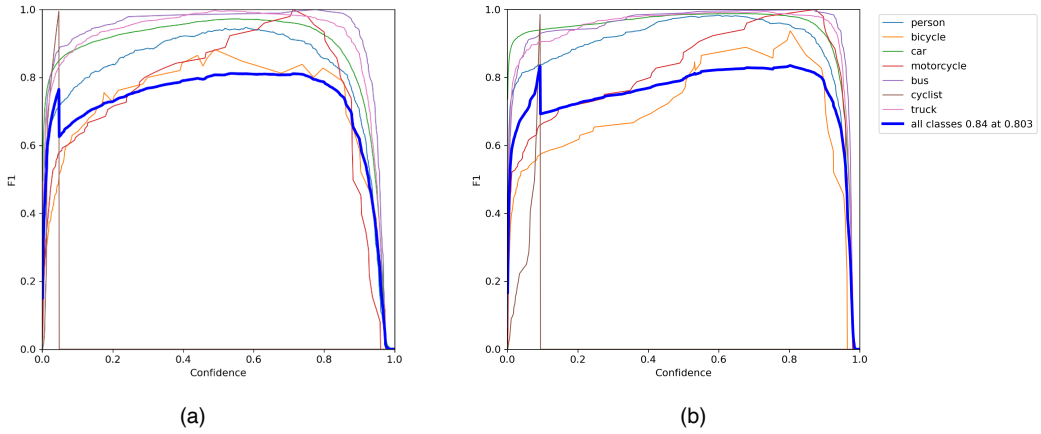

Figure 12. F1 curves for training our proposed YOLOv5 in small- and large-sized target, where the performance of the suggested model saturates gradually towards larger thresholds, indicating a more robust representation as (**a**) and (**b**) show.

The enhanced method was employed to generate PR curves before and after training on initial and improved datasets. PR curves capture the area formed by precision (P) and recall (R). Recall falls and accuracy rises with confidence. Determining the confidence level that optimizes F1 across all classes is the goal. The confidence threshold has to be raised when the goal is to reduce the number of FPs. The confidence level may be decreased if the goal is to find every potential item and producing FPs is not crucial. In Figure 13 and section (a), a distinct trend is observed where the areas of the three plots sequentially expand. A confidence of 0.95 and 0.81 has been identified to maximize accuracy and recall, corresponding to an F1 value of 0.883. One can note that when the IoU threshold rises, precision rises while recall falls. When the precisions and calls in these situations approach a particular threshold (0.5), known as the balancing point, all of the positive predictions are, in fact, true positives.

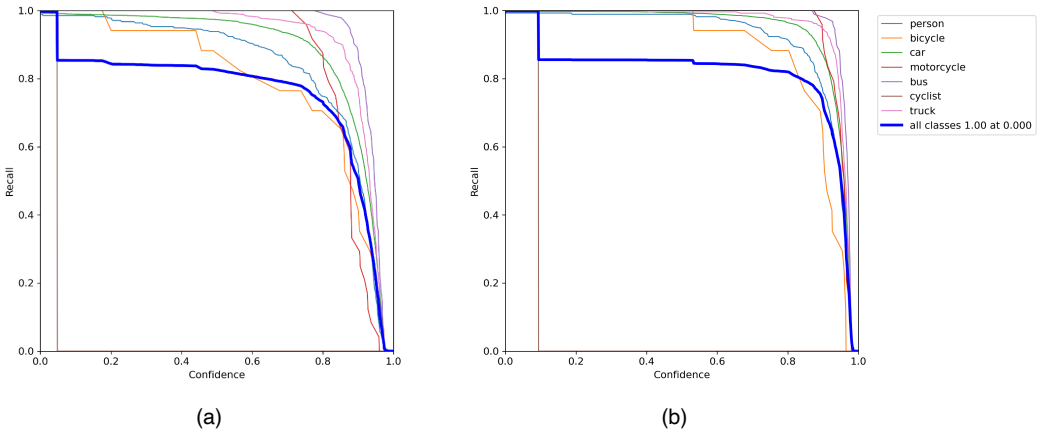

Figure 13. Recall curves for training our proposed YOLOv5 in (**a**) small- and (**b**) large-sized targets. The recall is 1 at a confidence of 0.

In the second training session, our enhanced YOLOv5 model was trained specifically for extra-large targets, following the default methodology applied in the initial training process. As a result, the default methodology, incorporating extra-large (x) models within our enhanced YOLOv5 algorithm, proved instrumental in achieving optimized results on the proposed dataset.

Thus, the experimental results demonstrate the effectiveness of the enhanced algorithm in object-detection tasks, particularly when configured for smaller target sizes. The Precision-Recall curves generated before and after training illustrate consistent improvements, with the AUCs expanding sequentially. It is worth noting that the use of

an extra-large target in our enhanced model, trained using default methodology, has further optimized results on the dataset. These findings point out the robustness and competence of the proposed approach in achieving high precision, recall, and accuracy in object-detection tasks.

5.3. Quantitative Evaluation

Utilizing crucial evaluation metrics, Section 5.3 substantiates the effectiveness of the proposed approach in target identification. In Table 4, our enhancement of YOLOv5x demonstrates improved performance in terms of mAP@0.5, mAP@0.95, and recall for recognizing large objects, achieving scores of 0.99, 0.90, and 0.99, respectively. These metrics significantly surpass the corresponding values for the baseline YOLOv5, which are 0.90, 0.69, and 0.85, as indicated in Table 3. Our proposed methodology achieves results comparable to the original YOLOv5, as evidenced by the visualized results in the test datasets shown in the visualized detection results. Notably, in comparison to the proposed YOLOv5 in all other detection aspects, the results demonstrate clear advantages in favor of our approach.

Table 4 also highlights that the enhanced methods outperform the benchmark in the state-of-the-art. Compared to the YOLOv5-baseline results in Table 3, mAP@0.95 improved by 24.9%, and mAP@0.5 improved by 14.4%. Both recall and precision improved by 17% and 6%, respectively. Consequently, the suggested detection head excels in retaining the features of smaller objects. Furthermore, involution efficiently enhances channel information, and the CBAM Block highlights important aspects while extracting them from the backbone.

5.4. Performance Comparison

In Section 5.4, a unified metric is presented that evenly weighs both precision and recall ratios; for the F1 score to increase, both recall and precision ratios must be higher. Figures 14 and 15 illustrate the tracking of metric curves for the model's training data using TensorBoard, both before and after the enhancement. Real-time visual inspection of the primary algorithm's performance is also demonstrated. The metrics of the model's performance are consistently evolving with increasing epochs, and it can be observed that precision and recall levels are gradually increasing in the mAP@0.5:0.95 group at 450 epochs.

The results of training a YOLOv5s model and an improved method on the DAWN dataset are depicted in Figures 14 and 15. The graphs present the evaluation metrics, including bounding box loss, mean target detection loss, and mean classification loss, for both the training and validation sets. In Figure 15, the enhanced models not only enhance detection accuracy and recall rates for small targets but also achieve a low final loss value with minimal fluctuations, indicating stable and robust learning.

Figure 11 illustrates the confusion matrices of the proposed method for a comparative analysis using the validation dataset, with the findings generated at an IoU of 0.5. The highest forecasts are indicated by the deepest blue shade in the color bar located on the right side, demonstrating a color spectrum ranging from 0 to the maximum predictions. Our proposed one-stage deep learning-based approach has made predictions on images containing cars, bicycles, and trucks in adverse weather conditions. Analyzing the objects within the confusion matrices for each class, the improved method identified more than 4.4% in cars, 80% in trucks, and 42% in bicycles compared to the YOLOv5 baseline. The new proposed method exhibits increased robustness in accurately identifying objects.

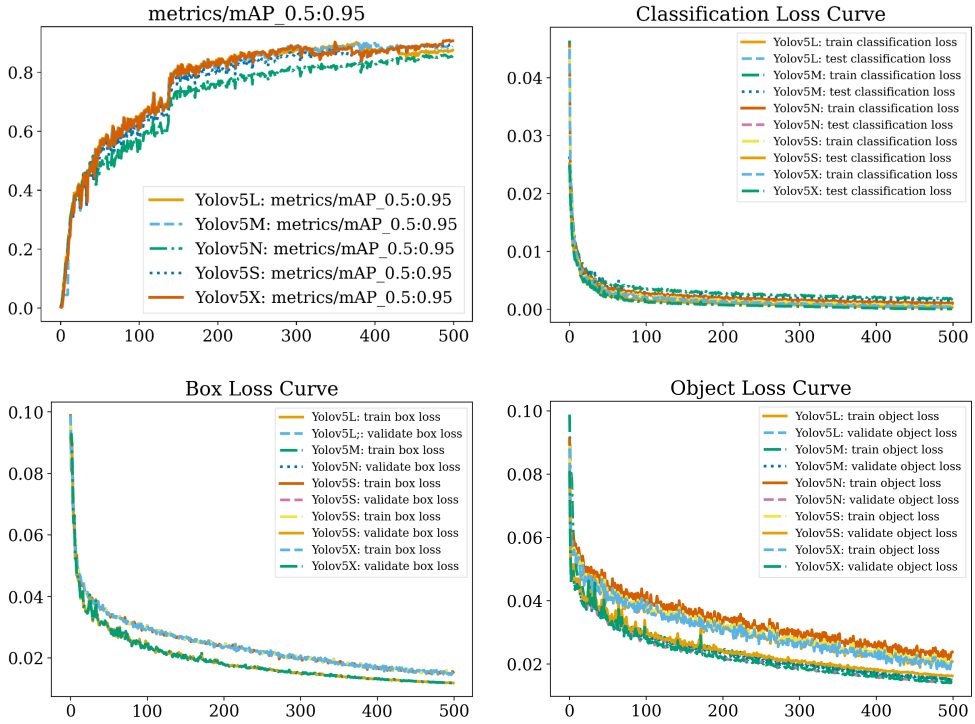

Figure 14. The results of training enhanced YOLOv5 for the extra large size model with the CBAM.

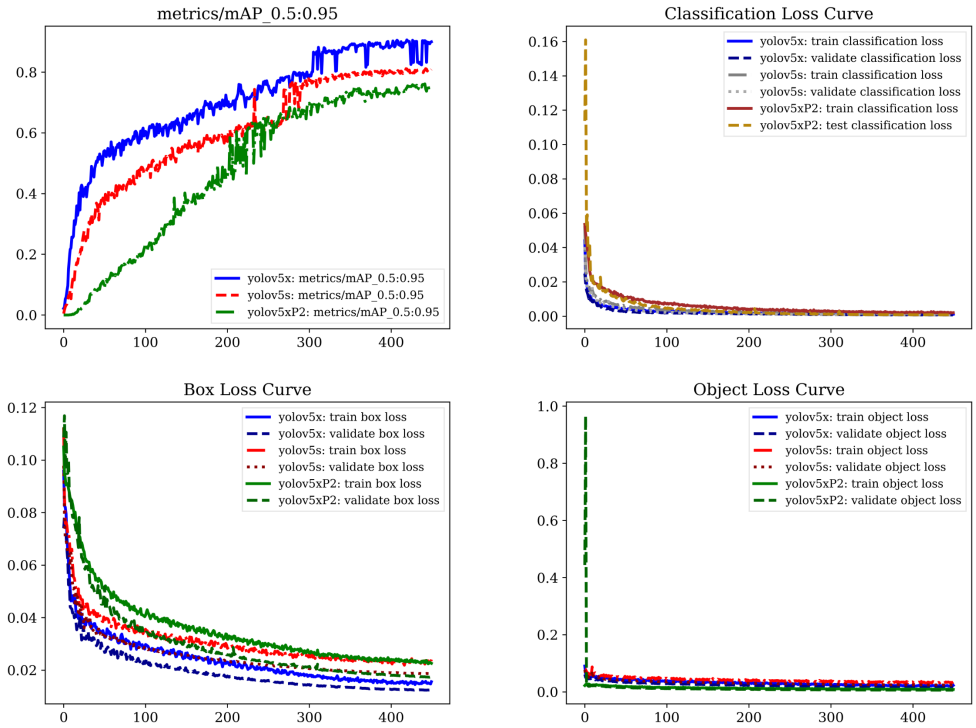

Figure 15. The results of training YOLOv5 for extra large size model with the CBAM.

5.5. Navigating the Trade-Off in Bounding-Box Accuracy

The evaluation of the COCO dataset follows standard metrics, as referenced in [51]. The AP measurements at various IoU thresholds, such as AP@0.5 and AP@0.95, which are the average over ten equidistant IoU thresholds from 0.5 to 0.9, are incorporated into our research. The relevant results for the bounding box-based object identification and mask-based instance segmentation formulations of the issue are shown in Table 5.

Figure 16 and Table 5 demonstrate the efficacy of five approaches across varying IoU thresholds. The F1 scores of all models exhibit a notable decline as the IoU threshold rises from 0.5 to 0.9. In particular, the baseline model decreased by 16%, and the enhanced model was reduced by 19%. In this study, statistical regression within the convolutional neural network's high-layer feature map is employed to forecast the bounding box for a region of interest (RoI). In this abstract feature map, each pixel in the original image correlates to a pixel block. In other words, a slight modification in the expected coordinates of the abstract feature map will result in a noticeable shift in the exact location within the original image.

Considering the results at the IoU 0.5, the enhanced method outperforms the baseline model by significant margins. In Table 5 for the precision assessment, our proposed upgrade demonstrates approximately a 7% improvement over the baseline model, yielding a precision of 0.92 for IoU 0.5. The enhanced YOLO model achieves an AP@0.7 of 0.91 when considering the performance values at IoU 0.70, providing a higher level of detection with bounding boxes that are considered correct. Still, in the table, when looking at the F1-Score evaluation, improved YOLO attains a much greater accuracy of 83.5% than the baseline model's 66.5%. In the case of recall scores at higher IoU thresholds, this observation may be expanded to include our improved solution: as the criteria for detection accuracy rise, our enhanced solutions continue to outperform the baseline method. Figure 16 displays the identical outcomes in Table 5.

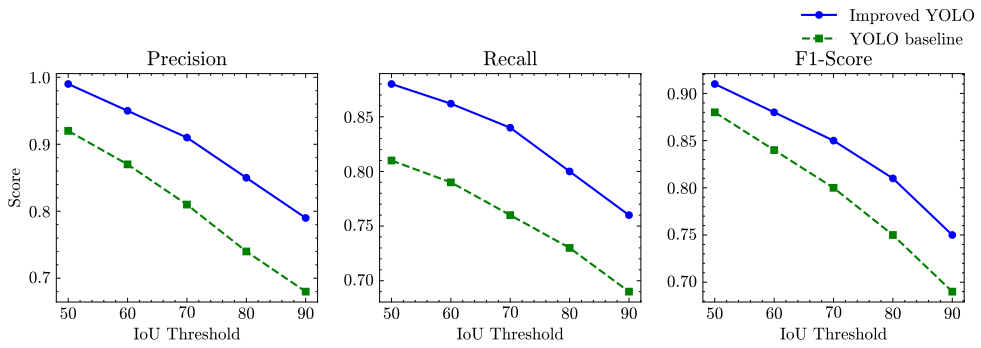

Figure 16. Comparison of improved and baseline models at different IoU threshold (%).

Table 5. Comparison performance at different IoU thresholds.

Methodologies	Metrics	IoU Threshold (%)				
		$AP_{0.5}$	$AP_{0.6}$	$AP_{0.7}$	$AP_{0.8}$	$AP_{0.9}$
YOLO baseline model	Precision	0.92	0.87	0.81	0.74	0.68
	F-Score	0.88	0.84	0.80	0.75	0.69
	Recall	0.81	0.79	0.76	0.73	0.69
YOLO enhanced model	Precision	0.99	0.95	0.91	0.85	0.79
	F-Score	0.91	0.88	0.85	0.81	0.75
	Recall	0.88	0.86	0.84	0.80	0.76

The findings indicate that the predictions of the suggested framework appear to be very accurate, as the average accuracy remains high as the IoU accuracy requirements increase, in the bottom part of Figure 16. This is crucial for our workflow, as accurate object prediction is required in the context of weather-adverse object recognition. Deep learning models may identify either an increased number of components with loose bounding boxes or a decreased number of items with proper bounding boxes [52]. However, the F1-score of YOLOv5 and enhanced YOLOv5 decreases as the IoU threshold increases.

5.6. Comparison Experiment of IoU Thresholds

As illustrated in Figure 17, a graph displaying average precision (%) versus threshold IoU is presented for different detection methods, including YOLO 5x, YOLO 5s, and YOLO5xP2, in the DAWN database. By reducing the IoU, more false negative samples are excluded, leading to a gradual improvement in the model's detection performance. The accuracy is higher when the IoU is 0.5, demonstrating a significant enhancement in the model's ability to detect objects. Therefore, YOLOv5XP2 exhibits superior robustness across three classes in the study.

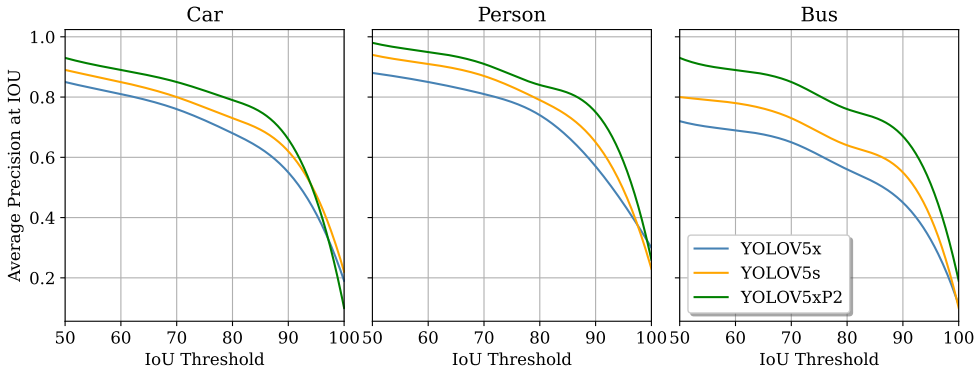

Figure 17. Average precision of improved models at different IoU thresholds (%).

5.7. Ablation Study

The sigmoid-weighted linear unit activation function (SiLU) [53] served as the foundation for the enhanced model. According to Liu et al. in [54], models with limited activation, such as ReLU, are amenable to quantization. However, models with unbounded activation functions, like SiLU or Hard-Swish, are not. Therefore, we retrained the models with ReLU activation. Changing the activation from SiLU to ReLU is observed to result in a decrease of approximately 1 to 2%. To position these models as embedded-friendly, we further quantified them. For instance, YOLOv5 for small targets with the ReLU function demonstrates that these models may be quantized with a minor drop of around 1.2% accuracy.

Specifically, post training quantization [55] is used to derive the findings below, instead of quantization aware training [56]. Figure 11 illustrates an example of inference results for row and column detection. The models performed well at lower IoU thresholds, with F1 scores of 93.4% and 94.7%, respectively. At the 90% IoU threshold, the graph showed encouraging results with F1 scores of 57.4% and 58.2%, respectively. Inference results for separate models are presented in Figures 12 and 13. These graphs demonstrate that individual models detect more than a combined model does.

We trained five different models and compared their performances in Table 6 to validate the contribution of the proposed algorithm module. According to the results, the components of the network slimming and attention module can increase the accuracy of the model by 2.15%. The model precision can be improved by 0.39% to 2.15% using the P2 Head. The accuracy of the model may be raised by 0.10% to 2.15% using the suggested enhanced components. In Section 3, it is evident that all optimization strategies mentioned notably enhance the mean average precision scores. It is worth noting that the marked performance enhancement of YOLOv5, specifically when equipped with a P2 head and attention module, particularly benefits small target models.

Table 6. Comparison results of ablation experiments on DAWN dataset.

Methodologies	Solutions	mAP@0.5	mAP@0.5:0.9
TPH-Slimming Pruned	Proposed method	0.94 (↑2.15)	0.81 (↑2.15)
YOLOv5x	Proposed method	0.75 (↑1.10)	0.99 (↑1.15)
YOLOv5s	Proposed method	0.95 (↑0.10)	0.81 (↑0.15)
YOLOv5s+ YOLOv5xP2CBAM	Proposed method	0.97 (↑2.15)	0.80 (↑2.15)
YOLOv5xP2+YOLOv5s	Proposed method	0.90 (↑0.39)	0.83 (↑0.05)

5.8. Detection Results Comparison

The outcomes on the test set of the baseline and enhanced models are depicted in Figure 18. For large targets, both methodologies can accurately identify objects. The recognition confidence of our proposed method, indicating that the improved YOLOv5 has enhanced ability in terms of foreground probability compared to YOLOv5, is significantly higher. In some scenes with dense targets, as shown in Figure 18c, the YOLOv5 algorithm has missed detection due to challenging weather conditions and instances of vehicle or pedestrian overlap and occlusion. However, the improved YOLOv5 still accurately identifies its target.

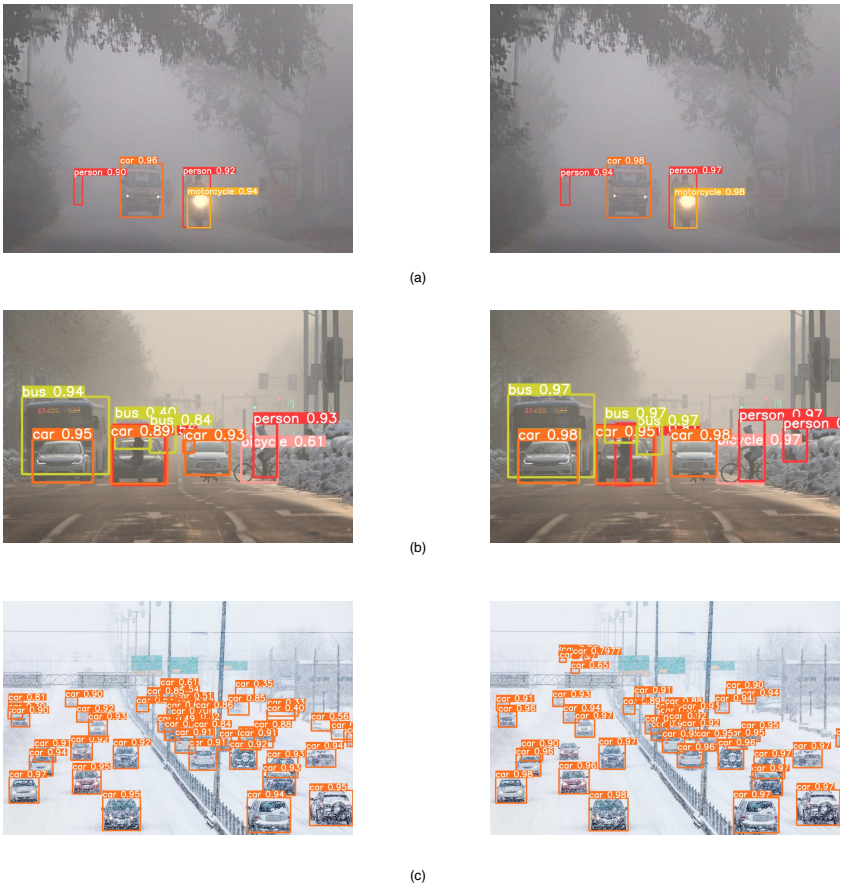

Figure 18. *Cont.*

(d)

(e)

Figure 18. The detection outcomes for each class include many epochs from the training and validation stages. The outcomes for YOLOv5 are on the left; on the right are the results of the upgraded model. In (**a**), the right hand side results showcase the effectiveness in achieving high-performance-inference while maintaining minimal computation utilization in foggy environment especially towards car and person. In (**b**), identifying pedestrians in sandy areas can significantly enhance safety, particularly in locations prone to vehicular traffic. In (**c**–**e**), the enhanced models exhibit enhanced performance in identifying objects inside the snow-covered landscape.

6. Conclusions

The proposed object detector aims to enhance the cutting-edge approach and capability for road targets in adverse weather conditions. This study explores the impacts and rationale behind various modifications to the architecture and model of the widely used one-stage object detector, YOLOv5.

The improvement to the YOLOv5-based method involves incorporating Swin transformers and CBAM, replacing the YOLOv5 loss function with the efficient IoU (EIoU) function, and integrating the YOLOv5 head with the TPH. Numerical results in this study showcase a mAP of 99.1% at an IoU of 0.5, surpassing the state-of-the-art result of 84.7% under the DAWN dataset. The proposed framework demonstrates robustness when compared across various performance metrics at different IoU thresholds.

Despite the progress made in this area, there are still limitations, especially due to the complex and dynamic nature of weather changes. The suggested framework improves safety, ranging from monitoring traffic flow to detecting potential hazards in industrial environments. In future work, we plan to assess the precision and processing time of model training on lightweight devices.

Author Contributions: Methodology, B.Z., M.S. and B.K.; Software, B.Z.; Validation, M.S., M.K. and B.K.; Formal analysis, M.S. and M.K.; Investigation, B.Z., M.S., M.K. and B.K.; Resources, B.K.; Data curation, B.Z.; Writing—original draft, B.Z.; Writing—review & editing, M.S., M.K. and B.K.; Supervision, B.K.; Project administration, B.K.; Funding acquisition, B.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) CREATE TRAVERSAL program.

Data Availability Statement: All data underlying the results are available as part of the article and no additional source data are required.

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

References

1. Taherifard, N.; Simsek, M.; Kantarci, B. Bridging connected vehicles with artificial intelligence for smart first responder services. In Proceedings of the 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Ottawa, ON, Canada, 11–14 November 2019; pp. 1–5.
2. Taherifard, N.; Simsek, M.; Lascelles, C.; Kantarci, B. Machine learning-driven event characterization under scarce vehicular sensing data. In Proceedings of the 2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Pisa, Italy, 14–16 September 2020; pp. 1–6.
3. Zhu, J.; Li, X.; Jin, P.; Xu, Q.; Sun, Z.; Song, X. MME-YOLO: Multi-Sensor Multi-Level Enhanced YOLO for Robust Vehicle Detection in Traffic Surveillance. *Sensors* **2021**, *21*, 27. [CrossRef] [PubMed]
4. Chen, Y.; Deng, C.; Sun, Q.; Wu, Z.; Zou, L.; Zhang, G.; Li, W. Lightweight Detection Methods for Insulator Self-Explosion Defects. *Sensors* **2024**, *24*, 290. [CrossRef]
5. Wang, T.; Zhai, Y.; Li, Y.; Wang, W.; Ye, G.; Jin, S. Insulator Defect Detection Based on ML-YOLOv5 Algorithm. *Sensors* **2024**, *24*, 204. [CrossRef]
6. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. *arXiv* **2018**, arXiv:1807.11164.
7. Lee, Y.; won Hwang, J.; Lee, S.; Bae, Y.; Park, J. An Energy and GPU-Computation Efficient Backbone Network for Real-Time Object Detection. *arXiv* **2019**, arXiv:1904.09730.
8. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *arXiv* **2019**, arXiv:1709.01507.
9. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. *arXiv* **2018**, arXiv:1807.06521.
10. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. *arXiv* **2020**, arXiv:1910.03151.
11. Yao, J.; Fan, X.; Li, B.; Qin, W. Adverse Weather Target Detection Algorithm Based on Adaptive Color Levels and Improved YOLOv5. *Sensors* **2022**, *22*, 8577. [CrossRef]
12. Khan, N.A.; Jhanjhi, N.; Brohi, S.N.; Usmani, R.S.A.; Nayyar, A. Smart traffic monitoring system using Unmanned Aerial Vehicles (UAVs). *Comput. Commun.* **2020**, *157*, 434–443. [CrossRef]
13. Lu, L.; Dai, F. Accurate road user localization in aerial images captured by unmanned aerial vehicles. *Autom. Constr.* **2024**, *158*, 105257. [CrossRef]
14. Kohli, P.; Chadha, A. Enabling Pedestrian Safety using Computer Vision Techniques: A Case Study of the 2018 Uber Inc. Self-driving Car Crash. *arXiv* **2018**, arXiv:1805.11815.
15. Li, X.; Cui, H.; Rizzo, J.; Wong, E.; Fang, Y. Cross-Safe: A Computer Vision-Based Approach to Make All Intersection-Related Pedestrian Signals Accessible for the Visually Impaired. In *Advances in Computer Vision—Proceedings of the 2019 Computer Vision Conference CVC*; Advances in Intelligent Systems and Computing; Arai, K., Kapoor, S., Eds.; Springer: Cham, Switzerland, 2020; pp. 132–146. [CrossRef]
16. Lu, L.; Dai, F. Automated visual surveying of vehicle heights to help measure the risk of overheight collisions using deep learning and view geometry. *Comput.-Aided Civ. Infrastruct. Eng.* **2022**, *38*, 194–210. [CrossRef]
17. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Lecture Notes in Computer Science*; Springer International Publishing: Chapel Hill, SA, USA, 2016; pp. 21–37. [CrossRef]
18. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. *arXiv* **2020**, arXiv:1911.09070.
19. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *arXiv* **2018**, arXiv:1708.02002.
20. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2016**, arXiv:1506.02640.
21. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. *arXiv* **2016**, arXiv:1612.08242.
22. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In *Lecture Notes in Computer Science*; Springer International Publishing: Beijing, China, 2014; pp. 346–361. [CrossRef]
24. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
25. Misra, D. Mish: A Self Regularized Non-Monotonic Activation Function. *arXiv* **2020**, arXiv:1908.08681.
26. Wang, W.; Wei, C.; Yang, W.; Liu, J. GLADNet: Low-Light Enhancement Network with Global Awareness. In Proceedings of the 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), Xi'an, China, 15–19 May 2018; pp. 751–755.
27. Loh, Y.P.; Liang, X.; Chan, C.S. Low-light image enhancement using Gaussian Process for features retrieval. *Signal Process. Image Commun.* **2019**, *74*, 175–190. [CrossRef]
28. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. *arXiv* **2017**, arXiv:1612.03144.
29. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.
30. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks. *CoRR* **2017**, arXiv:1703.06211.

31. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features from Cheap Operations. *CoRR* **2019**, arXiv:1911.11907.
32. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *CoRR* **2019**, arXiv:1905.11946.
33. Li, B.; Wu, B.; Su, J.; Wang, G.; Lin, L. EagleEye: Fast Sub-net Evaluation for Efficient Neural Network Pruning. *arXiv* **2020**, arXiv:2007.02491.
34. Cui, C.; Gao, T.; Wei, S.; Du, Y.; Guo, R.; Dong, S.; Lu, B.; Zhou, Y.; Lv, X.; Liu, Q.; et al. PP-LCNet: A Lightweight CPU Convolutional Neural Network. *CoRR* **2021**, arXiv:2109.15099.
35. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. *CoRR* **2021**, arXiv:2108.11539.
36. Li, Y.; Shen, M.; Ma, J.; Ren, Y.; Zhao, M.; Zhang, Q.; Gong, R.; Yu, F.; Yan, J. MQBench: Towards Reproducible and Deployable Model Quantization Benchmark. *arXiv* **2021**, arXiv:2111.03759.
37. Ding, H.; Pu, J.; Hu, C. TinyNeuralNetwork: An Efficient Deep Learning Model Compression Framework. *arXiv*, **2021**. Available online: https://github.com/alibaba/TinyNeuralNetwork (accessed on 1 September 2022).
38. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *arXiv* **2021**, arXiv:2103.14030.
39. Ultralytics. YOLOv5: A State-of-the-Art Real-Time Object Detection System. 2021. Available online: https://docs.ultralytics.com (accessed on 9 January 2023).
40. Shen, F.; Zeng, G. Weighted Residuals for Very Deep Networks. *arXiv* **2016**, arXiv:1605.08831.
41. Anzaroot, S.; Passos, A.; Belanger, D.; McCallum, A. Learning Soft Linear Constraints with Application to Citation Field Extraction. *arXiv* **2014**, arXiv:1403.1349.
42. Lang, X.; Ren, Z.; Wan, D.; Zhang, Y.; Shu, S. MR-YOLO: An Improved YOLOv5 Network for Detecting Magnetic Ring Surface Defects. *Sensors* **2022**, *22*, 9897. [CrossRef]
43. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**, arXiv:2010.11929.
44. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2023**, arXiv:1706.03762.
45. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and tracking meet drones challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 7380–7399. [CrossRef] [PubMed]
46. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
47. Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; Zhang, C. Learning Efficient Convolutional Networks through Network Slimming. *arXiv* **2017**, arXiv:1708.06519.
48. Kenk, M.A.; Hassaballah, M. DAWN: Vehicle Detection in Adverse Weather Nature Dataset. *arXiv* **2020**, arXiv:2008.05402.
49. Yuan, G.X.; Chang, K.W.; Hsieh, C.J.; Lin, C.J. A Comparison of Optimization Methods and Software for Large-scale L1-regularized Linear Classification. *J. Mach. Learn. Res.* **2010**, *11*, 3183–3234.
50. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Müller, A.; Nothman, J.; Louppe, G.; et al. Scikit-learn: Machine Learning in Python. *arXiv* **2018**, arXiv:1201.0490.
51. Lin, T.Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context. *arXiv* **2015**, arXiv:1405.0312.
52. Kumar, A.; Zhang, Z.J.; Lyu, H. Object detection in real time based on improved single shot multi-box detector algorithm. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 204. [CrossRef]
53. Elfwing, S.; Uchibe, E.; Doya, K. Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning. *arXiv* **2017**, arXiv:1702.03118.
54. Liu, R. Higher Accuracy on Vision Models with EfficientNet-Lite. 2020. Available online: https://blog.tensorflow.org/2020/03/higher-accuracy-on-vision-models-with-efficientnet-lite.html (accessed on 16 March 2023).
55. Liu, Z.; Wang, Y.; Han, K.; Ma, S.; Gao, W. Post-Training Quantization for Vision Transformer. *arXiv* **2021**, arXiv:2106.14156.
56. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. *arXiv* **2017**, arXiv:1712.05877.