electronics

MDPI

Article

Object and Pedestrian Detection on Road in Foggy Weather Conditions by Hyperparameterized YOLOv8 Model

Ahmad Esmaeil Abbasi, Agostino Marcello Mangini * and Maria Pia Fanti *

Department of Electrical and Information Engineering, Polytechnic University of Bari, 70125 Bari, Italy; a.esmaeilabbasi@phd.poliba.it

* Correspondence: agostinomarcello.mangini@poliba.it (A.M.M.); mariapia.fanti@poliba.it (M.P.F.)

Abstract: Connected cooperative and automated (CAM) vehicles and self-driving cars need to achieve robust and accurate environment understanding. With this aim, they are usually equipped with sensors and adopt multiple sensing strategies, also fused among them to exploit their complementary properties. In recent years, artificial intelligence such as machine learning- and deep learning-based approaches have been applied for object and pedestrian detection and prediction reliability quantification. This paper proposes a procedure based on the YOLOv8 (You Only Look Once) method to discover objects on the roads such as cars, traffic lights, pedestrians and street signs in foggy weather conditions. In particular, YOLOv8 is a recent release of YOLO, a popular neural network model used for object detection and image classification. The obtained model is applied to a dataset including about 4000 foggy road images and the object detection accuracy is improved by changing hyperparameters such as epochs, batch size and augmentation methods. To achieve good accuracy and few errors in detecting objects in the images, the hyperparameters are optimized by four different methods, and different metrics are considered, namely accuracy factor, precision, recall, precision–recall and loss.

Keywords: object detection; pedestrian detection; deep learning; machine learning; bounding box; YOLOv8 model

Citation: Esmaeil Abbasi, A.; Mangini, A.M.; Fanti, M.P. Object and Pedestrian Detection on Road in Foggy Weather Conditions by Hyperparameterized YOLOv8 Model. *Electronics* **2024**, *13*, 3661. https://doi.org/10.3390/electronics13183661

Academic Editors: Yue Wu and Beiwen Li

Received: 6 July 2024
Revised: 4 September 2024
Accepted: 12 September 2024
Published: 14 September 2024

1. Introduction

Autonomous systems are eye-catching technologies that are being used nowadays by some industries, companies, and individuals. Autonomous systems can be used in various fields with the help of object detection technologies in industries, medicine, departure gates, and autonomous vehicles [1]. Recently, technologies including machine learning (ML) and deep learning (DL) have played an important role in research [2] and are adopted in autonomous systems to detect objects on streets [3]. Autonomous vehicle companies use sensors and object detection models to detect objects on roads in adverse weather conditions by detecting people and cars with the help of sensors like Lidar, ultrasonics, and GPS. A lack of information on roads in autonomous vehicles leads to more crashes in worse weather conditions as well. Self-driving cars use these technologies to prevent accidents on roads and save human lives better than skilled drivers [3]. Detecting objects in images characterized by normal weather conditions is quite simple. However, object detection becomes more complicated in foggy, rainy, snowy, and other adverse weather conditions. Indeed, one of the main detection problems is identifying an object in adverse weather conditions. Some objects can be detected by standard models, but significant challenges arise in detecting objects in foggy conditions. When the fog is thick, road images suffer from reduced visibility, quality, contrast, color, and sharpness.

Building on these challenges in adverse weather conditions, our research focuses on a critical aspect of autonomous driving. The main target of this study is detecting objects in adverse weather conditions which can also be used in self-driving vehicles without the help

of humans. In this paper, a procedure is proposed to solve the object detection problem in the presence of foggy conditions. The aim is to reach a higher object detection rate with respect to the other works presented in the literature of the sector. Some previous research tried to use various models to detect objects in images characterized by adverse weather conditions [4]. However, they did not cover and detect all the main items on the street, such as cars, pedestrians, traffic lights and street signs. The proposed procedure attempts to narrow this gap by using a YOLOv8 model to detect crucial objects in foggy weather conditions for autonomous driving.

ML and DL are fed by raw materials, such as texts, databases and images [5]. Naturally, in object detection, the raw material is represented by images. Some of the more efficient DL models are CNNs (convolutional neural networks), Fast-RCNNs (Fast Region-based Convolutional Neural Networks) and YOLO (You Only Look Once) [6]. The eighth version of YOLO (YOLOv8) is the fastest DL model to recognize objects in images [7,8]. In [9], experts utilized a modified YOLOv7 called Ghost-YOLOv7, which enhances the feature extraction capability and detects vehicles in normal weather conditions.

While these models have shown promise in various applications, researchers have adapted and combined them to address specific challenges in object detection. Several studies have explored different approaches to tackle object detection in various weather conditions and scenarios. In [4], the authors present a method based on cooperation between YOLOv3 and a Hue–Saturation–Value (HSV) color space to detect vehicles on streets in rainy weather conditions. The proposed method compares image frames in good and rainy weather conditions captured by cameras placed in specific locations. In this context, the method fixes items, rain droplets, and vehicles by cameras. In [10], a method based on Faster-RCNN is proposed to recognize traffic lights on roads. The method is characterized by reduced detection time due to AlexNet, a trained linear support vector machine.

Paper [11] presents a method to detect vehicles, pedestrians and etc by using images taken from the sky by drones. The authors used and compared various models such as YOLOv3, YOLOv4, R-CNN and Faster-RCNN to achieve high accuracy and a fast detection rate. In [12], the authors employ an R-CNN model to detect street signs with the help of DL. They use R-CNN to extract texts from the street signs to achieve a high detection rate. In other works, researchers combine well-known models and some pre-trained models. In [5], the authors use a modified YOLOv3 model to detect multiple objects with the help of a Common Objects in Context (COCO) pre-trained dataset. Unfortunately, their model cannot detect all objects in images.

In [13], a model was investigated to detect traffic lights on roads. The authors use the color segmentation and shape property detection of different models integrated into their model. Paper [6] provides a method based on a CNN to detect traffic lights in low and high lighting conditions. The model is characterized by two output layers: the first layer, having one output value, is used for the binary classification task of determining whether a sample contains traffic lights; the second layer has four output values and is used for the regression task of traffic light localization to determine the upper-left point of a corresponding bounding box and its width and height. In [14], a model is presented based on modified Faster-RCNN model, in its model layers. The authors added a deconvolutional layer to the output and called it the DR-CNN model. The proposed model detects traffic signs on the street and categorizes them into three groups, including small, medium and large objects.

Despite these advancements, there remains a significant gap in the literature regarding comprehensive object detection in foggy conditions. However, none of the mentioned papers can detect cars, pedestrians, traffic lights or street signs in foggy weather conditions. In other research [1,13,15–30], some researchers tried to detect two or three object classes in their research, and their results in some cases are not excellent.

This paper proposes a new methodology to build a model that is able to detect objects in adverse weather conditions and in particular in the presence of foggy weather. The proposed approach can be efficiently used in autonomous vehicles. The aim is to reach

a higher object detection rate with respect to the works presented in the related literature. The methodology can be utilized to detect cars, pedestrians, traffic lights and street signs by using YOLOv8 without modifying it or adding relevant filters. On the other hand, we select some appropriate hyperparameters by trial and error. More precisely, by deep learning application, the hyperparameters are set or configurations are often established before the learning process begins. Some examples of frequent hyperparameters are the following: batch size, epoch, learning rate, augmentation and optimizer.

The new contribution of the paper is the following. While the proposed method is based on the existing YOLOv8 model, the core innovation lies in its application and optimization for object detection in foggy street images, a challenging and less-explored domain. Our approach includes several novel aspects such as image preprocessing, where we adapt the image quality and size to better suit the training process, crucial for enhancing performance in varied real-world scenarios. We conducted thorough hyperparameter optimization, exploring their impacts in detail to finely tune our model for detecting objects in foggy conditions. Additionally, we utilized brightness augmentation, which significantly improved detection rates under foggy conditions. Our study systematically evaluated this augmentation, providing insights into its benefits and limitations. We also employed different variants of the YOLOv8 model (small and large) to understand performance differences, highlighting trade-offs between model complexity and performance. To address concerns about detailed discussion, we expanded the paper to include a comprehensive analysis of brightness augmentation's impact on detection accuracy, with quantitative results and qualitative examples. The paper also compares the small and large YOLOv8 models, highlighting their respective advantages and suitability for different applications.

The paper is structured as follows. Section 2 presents the state of the art of object detection models, while Section 3 introduces the problem statement. In Section 4, the proposed methodology is described in detail and, in Section 5, the efficacy of the object detection model is shown by graphs, figures and a wide discussion. Finally, Section 6 gives the conclusion and discusses future works.

2. Related Works

Deep learning has transformed object detection, enabling the precise identification of essential elements like cars, crossroads, traffic lights and signs. Among deep learning models, the strength of YOLO is its single neural network, predicting bounding boxes and class probabilities directly from images, ensuring both speed and accuracy. Its unified architecture streamlines detection, classification and localization tasks, making it ideal for applications needing swift decisions, such as autonomous vehicles and traffic management. YOLO continues to drive object detection forward, enhancing safety and efficiency, in our whole world. In paper [26], the authors used the YOLOv7 model and modified it to detect only people on roads. They used the object detection algorithm to identify people in foggy conditions; the dark channel de-fogging algorithm was incorporated, along with four layers of down-sampling that decreased the number of pixels in each dimension and led to a reduced amount of data. Then they used up-sampling to increase the number of pixels in each dimension to accelerate the de-fogging process. To maintain accuracy in complex environments, the ECA module and an additional detection head were integrated into the YOLOv7 network architecture to enhance object classification and detection.

In [18], the authors employed the YOLOv5 architecture as the foundation and adapted it into two distinct models: the student and teacher models. The student model underwent updates through backpropagation, while the teacher model's weights were adjusted using the Exponential Moving Average (EMA) of the student model. Effectively, the teacher model aggregated insights from multiple instances of the student model over time. Their approach involved alternating between training the student model on a supervised source domain and a pseudo-supervised target domain. Simultaneously, the teacher model generated pseudo labels for the target domain by updating them through the EMA, leveraging the outputs of the student model. Additionally, they proposed the use of a cross-attention

strategy transformer to facilitate the alignment of features between the source and target domains through knowledge distillation. Their model was able to detect people, cars, bikes, trains and motorbikes.

Also, the YOLO model can use a pre-trained dataset to be trained in order to identify some objects. The YOLOv3 model was used in [7] for car detection. They used a pre-trained COCO dataset as input images in their study. They used COCO to focus on detecting cars on roads. We also used COCO on the first try, but the traffic signs and traffic lights had not been annotated when we used COCO. As a result, we decided to create a model to detect those objects with the help of the YOLOv8 model.

In [14], the authors benefited from a deconvolution region-based convolutional neural network (DR-CNN) to cope with small objects in large images. Since, when using R-CNN, there are a lot of layers in the neural network, detecting small objects will be complex. Firstly, they added a deconvolution layer and a normalization layer to the output of the convolution layer. They tried to detect traffic lights in their study.

In [20], the Faster-RCNN network architecture was applied with feature concatenation and hard negative mining strategies to detect pedestrians on streets. They leveraged a specific classifier to improve the pedestrian detection rate in their study, then trained their model on the Daimler pedestrian dataset. In the next step, they utilized a hard negative mining approach to classify samples with classifiers and use the misclassified samples as negative samples to continue training the classifier.

2.1. Data to Be Used

The first step and important key in object detection is how data are collected. The data play a crucial role in object detection, and texts, images and numbers can be used as raw material in ML and DL technologies [5]. For instance, in autonomous vehicles, we need to detect the objects surrounding a self-driving vehicle by Lidar, sensors and object detection technologies [14]. Generally, two approaches for collecting information on the street are mainly used: manual and semi-automated methods, respectively [3].

In the manual method, all images are collected by humans. However, this method takes too much time and the results are inefficient.

Conversely, in the semi-automated approach, the images are collected on roads by a vehicle equipped with a high-quality dashboard camera recorder. After the images are extracted frame by frame, they are processed by a computer and training and validation datasets are obtained [3,29].

In this paper, we use YOLOv8 and the Roboflow (https://roboflow.com/, accessed on 11 September 2024) annotation tool as well as a dataset that includes about 4000 foggy street images which were collected by a semi-automated approach. Moreover, the object detection results about these 4000 foggy street images are evaluated by the following metrics: precision, recall, precision–recall, accuracy and loss [5].

2.2. Object Detection Models and Metrics

Several models are used by researchers to detect objects on the street. Some scientists use Fast-RCNNs which is another popular model to detect objects on roads. In [14], the authors train and validate a DR-CNN model powered with a deconvolution layer to detect traffic light shapes and positions using a pre-trained dataset consisting of 165,000 training images ("2015 train") and 81.000 validation images ("2015 val").

In DL technology, an iteration is an important factor that consists of the following multiple steps:

- Forward Pass. During an iteration, a set of training data is passed forward via the neural network. Each sample in the batch goes through a series of calculations defined by the network's architecture, where weights and biases are applied to the input data to generate forecasts.
- Loss Computation. After the forward pass, the predictions made by the neural network are compared to the actual targets (ground truth). This comparison is performed using

a loss function, which measures how well the network performed on that particular batch of data.

- Backward Pass (backpropagation). The calculated loss is then used to adjust the model's internal parameters (weights and biases) in a process called backpropagation. The goal is to minimize the loss by adjusting these parameters. This step involves computing the gradients of the loss function for each parameter in the network.
- Optimizer Update. The optimizer algorithm, such as stochastic gradient descent (SGD), Adam or Root Mean Squared Propagation (RMSprop), uses gradients to update the model's weights and biases. The optimizer determines the direction and magnitude of the updates applied to the parameters to reduce the loss.
- Repeating for Multiple Iterations. This process is repeated multiple times, incorporating new batches of data each time until the whole dataset is used (completion of an epoch) or until a predetermined number of iterations is reached. The steps involved are forward pass, loss computation, backward pass and optimizer update.

In [10], traffic lights are detected by 5000 iterations performed by a Fast-RCNN model. In [23], 120,000 iterations are used to detect pedestrians by the Fast-RCNN model and the Regional Proposal Network feature.

In [20], the authors present a combination of the Fast-RCNN model with Online Hard Example Mining (OHEM) to detect objects like traffic signs on roads with a real-time method. The proposed method has some advantages, such as improving training focus, since OHEM helps focus the training process on difficult examples, which are often crucial for the model's improvement. By prioritizing these hard examples, the model can learn from the most informative instances, potentially improving its overall performance. In paper [21], the status of a traffic light is recognized by DL technology and YOLOv3. The model helps ensure that traffic light statuses such as red, green and yellow can be recognized by the proposed model. Paper [26] modifies a model based on YOLOv7 to detect pedestrians on foggy streets images with the help of a de-fogging filter. In [29], the authors improve text detection on roads. In particular, they work on identified texts on Malaysia street images and extract texts from the images by using a histogram of oriented gradients and an Artificial Neural Network (ANN). First, a histogram equalization is performed on the image frame to improve contrast. Second, the blue channel of the image is extracted and noise is removed from it (salt and pepper noise removal and median filtering). Finally, the authors convert texts into voices that can be used by disabled people or smart devices such as smart cars. Paper [22] detects traffic lights by using Global Positioning System (GPS) technology with the help of sensors and a ground-truth representation method, then converts images that are taken from the previous step and changes them to gray images. In the next level, texts are extracted from the traffic sign gray images. The result is an increased accuracy and reduced errors.

Up to now, the presented papers aim to identify a single class of objects on the road. A study has been performed to detect, by the same model, three classes including cars, traffic lights and pedestrians. However, the authors use unreal weather conditions in images, because they modify the images with the help of foggy weather simulation, filter or dedicated algorithms [18]. As a model, Roboflow (https://roboflow.com/, accessed on 11 September 2024) is used for image annotation and YOLOv5 to detect objects on streets. The result in precision and recall metrics for the three classes are not ideal [3,8]. In our study, we investigate the three same classes and street signs as a fourth class.

Others works study different objects to be detected like bikes. They analyze clear images detecting cars and pedestrians as well as bikes using a Fisheye8K camera and four different versions of YOLO, namely YOLOv5, YOLOR, YOLOv7 and YOLOv8.x, at four times during a day: morning (sunrise), afternoon (sunny), evening (sunset) and night [19].

Differently from paper [19], we use a dataset of dashboard camera images characterized by fog and the results are better than the results obtained in [19] in mAP50 and precision metrics. Paper [7] uses YOLOv3 with the help of a pre-trained COCO dataset in

normal weather, although with high-resolution pixel images. Unfortunately, the results and metrics are not good for all the three classes.

In [4], the authors use DL and combine it with hardware same as Raspberry PI in order to detect objects on roads in real time using wearable items such as a hat. They analyze some images before and after fog, rain and snow to detect better objects by comparing old and new images in the same location.

Some metrics are used in the mentioned models such as epoch, batch size, augmentation, optimizer, learning rate, etc. An epoch is one complete run through the training dataset; training for more epochs allows the model to learn more from the data, while training for too many epochs may result in overfitting. A larger batch size, on the other hand, may speed up computing but result in less generalization. This parameter scales the size of the updates to the model weights. A high learning rate may cause the model to overshoot the optimal solution, while a low learning rate may slow convergence. Augmentation involves applying random modifications to the training data, like rotations, flips, zooms or shifts. It helps the model generalize better by providing variations of the same data. Augmentation hyperparameters include the type and intensity of the applied transformations. Optimizer like Adam, SGD (stochastic gradient descent), and Root Mean Squared Propagation (RMSprop) are used to update the model parameters during training. Each optimizer has its own set of hyperparameters such as momentum, decay rates, etc.

In this paper, the epoch, batch size and augmentation (brightness augmentation) hyperparameters, and other hyperparameters be in default mode, are optimized. While the used optimizers in hyperparameter are the same as YOLOv8 auto mode and the learning rate remains constant. We demonstrate that the model takes a short time to achieve the detection result and its results are more accurate than other models that were mentioned above. Some researchers investigate the Fast-RCNN model to detect multi-class objects where it is hard to extract features and classification. However, the studies show a high computation time cost. Conversely, we use the recent version YOLOv8 that is about seven times faster than the Fast-RCNN model. Moreover, we train the proposed model by natural and real images without modifying them by algorithms or filters.

2.3. Object Detection Background

Detecting objects is performed by two classes of methods on roads: active and passive methods. The first method uses laser-equipped sensors (Lidar), while the second one uses cameras [15]. Models use the obtained images to train procedures in both active and passive methods.

Object detection has two different stages: the first stage compares images by training detection based on the whole images; the second stage detects objects from some parts of an image and trains detection based on region proposals. Lights, noise and low-quality textures can have negative effects on the images of a model.

Recognizing objects, plays an important role in autonomous vehicles and can lead to the saving of human lives. Approximately 700 people died in road accidents, which can be reduced if we can use modern technologies such as ML and DL to detect objects on streets and prevent collisions [15].

One of the most important aspects of image annotation is that the images be clear, and it is preferable to use colorful images rather than black and white images to train a model and detect objects. Moreover, if a specific dataset is not available, Google Street View allows specific images to be collected on the street [16].

Finally, drones and satellites can be used to help an autonomous vehicle during its trip. Some papers use drones to detect vehicles and pedestrians via images [1,11]. Other papers benefit from satellite images to improve object detection [30].

3. Problem Description

Detecting objects on roads becomes increasingly challenging in adverse weather conditions like rain, fog and snow. The dataset used consists of re-sized images captured

in foggy weather. Researchers attempted to identify objects such as cars, pedestrians, traffic lights and signs in these images despite several issues like noise and poor textures caused by foggy conditions. However, detecting objects in heavily foggy weather remains a significant challenge for computer models and even human observers [28]. This difficulty is particularly crucial in the context of autonomous vehicles, where accurate object detection in real-time environments is critical for safe navigation [15].

Enhancing object detection accuracy in road images, particularly in foggy conditions, is the primary goal of the proposed model. Indeed, successfully detecting objects on roads under foggy conditions could significantly enhance the control of autonomous systems, reduce accidents, improve traffic management, increase street safety and ultimately, save lives [10]. In this context, addressing issues related to image clarity and brightness, and using methodologies in ML and DL are essential for effective learning and object detection [8,9,12].

3.1. Methodology

In ML and DL technologies, the most important factors are the dataset and the hardware, particularly the Graphic Processing Unit (GPU). About 4000 foggy road images in Zurich—that were downloaded from the people.ee.ethz.ch website—were used as a dataset to create the proposed detection model. Benefiting from the Google Hardware Service called Google Colab [2], a powerful computer with high resources, we analyzed these images. The hardware resources that have been dedicated to the proposed model by Google Colab include a GPU A100 (40 GB of memory), 18 GB of memory, 2 CPUs and an 80 GB hard disk. The third important factor is that the images need to be bounding boxed or annotated by a tool or software as well, and all objects in the images must be marked [28]. Thus, we use Roboflow annotation tools online.

Before, we used Roboflow based on the pre-trained COCO model but it failed to accurately recognize objects within figures. For example, COCO could not detect clearly pedestrians and all street signs. By Roboflow, all desired objects will be annotated by a frame around the objects. The colorful frame will show the positions of the desired objects (cars, pedestrians, traffic lights and traffic signs) in road images. There are some challenges in marking the objects in the dataset. It is crucial to mark the objects correctly. The frames around the objects will have a direct connection to detect the objects better in the training process. All objects that will be annotated in the images will be considered as a class as well. In the study, we have four classes (cars, pedestrians, traffic lights and traffic signs).

This research focuses on object detection in foggy weather conditions using real foggy images and the YOLOv8 model. Unlike previous studies that rely on digitally altered images, this approach uses actual foggy pictures and introduces a novel preprocessing step of reducing the image size and quality before training. This method not only addresses the challenge of working with low-quality datasets but also enhances the model's robustness. This study explores different variants of YOLOv8 (small and large) and investigates the impact of brightness augmentation on detection accuracy.

The research contributes several innovations to the field, including a detailed analysis of hyperparameter optimization for foggy conditions, a systematic evaluation of brightness augmentation (varying from −25% to +25%) and a comparative study of YOLOv8 model variants. These aspects collectively address the challenges of object detection in adverse weather conditions and provide valuable insights for future research and practical applications. The expanded discussion in the paper offers both quantitative results and qualitative examples, demonstrating the method's effectiveness and its potential limitations, particularly in scenarios involving unclear images.

For this reason, we decided to create a proposed model by training a model to identify the objects in street images. Initially, we reduced raw image quality and size by 25% and 50%, respectively. This led to having high accuracy and improved outcomes even with lower image quality. Subsequently, we checked the images about the presence of objects. Using Roboflow software (https://roboflow.com/, accessed on 11 September 2024), any

re-sized image containing objects was boxed and allocated to our training folder for the proposed model. As a third phase, we employed YOLOv8 to train the proposed model for object detection in the re-sized images. Moreover, we explored four different sub-methods to find a good procedure in a model for detecting objects in foggy images. Since YOLOv8 is characterized by hyperparameters, we decided to utilize four different categories based on the hyperparameters in four sub-methods. As result, a general methodology is introduced and illustrated in Figure 1. In some articles, experts tried to customize some hidden layers in the YOLOv8 network to improve the accuracy of their model to detect objects better in image datasets. However, we did not use a YOLOv8 custom model with different hidden layers for identifying objects in the images, because it requires too much time for detection [8,25]. We benefited from the default hidden layers of YOLOv8 and changing the hyperparameters.

Based on Figure 1, the detection methodology consists of five steps that are described in detail in the following.

- Step 1 (Decreasing the image quality and size). The quality and size of all images are reduced to 25% and 50%, respectively.
- Step 2 (Identifying and marking objects in images). Some objects in the images such as cars, pedestrians, traffic lights and signs are identified. If some specific objects are identified in an image, they are annotated or marked by a tool such as Labelbox or Roboflow, and the image is stored in a training folder. Otherwise, if no object is present in an image, it is stored into the training folder directly.
- Step 3 (Image extraction for model validation). Twenty percent of the total image dataset is moved into a validation folder.
- Step 4 (Training). Four different methods with various configurations, characterized by different epochs and batch sizes, and with or without augmentation, are used to train the model to better detect the objects in the images.
- Step 5 (Outcomes). The results for each various approaches can be seen in the figures.

Raw images for training a model in deep learning lead to spending more time compared to compressed and reduced image data based on size and resolution. Images with high quality and size required more memory and a deeper network. The high quality and size of image resources result in a deep learning model selecting unnecessary details as well. But, in some studies related to human health, image details will be mandatory. For instance, in deep learning studies on tumors or cancers, the image details are crucial for training a model to detect a tumor or cancer from the input images. But, in the proposed model, the desired objects are placed on roads. They can be detected by the proposed model easily and do not need to have more detail.

In method 1 we started to use a small version of the YOLOv8 as the first hyperparameter metric, since it has fewer hidden layers than a large version of the YOLOv8. Also, the training time for the small version (YOLOv8S) was less than for the large version (YOLOv8L). As a second hyperparameter, we used 100 epochs in methods 1, 2 and 3 in the study. However, we utilized 150 epochs in method 4 to achieve better and more accurate outcomes. There is a direct relation between the number of epochs and object detection outcomes. Generally, by increasing epochs, an object detection model will be learned better. Also, the number of epochs can be changed and it depends on the model complexity, dataset size and task difficulty. As the third hyperparameter metric is augmentation, we do not use brittleness augmentation in method 1. Its range was between −25% and +25% in the image dataset. By decreasing and increasing brightness augmentation, where objects on roads are placed in the darkest or the lightest area in image dataset, the objects will be identified better. As the fourth hyperparameter metric, we leveraged batch size 16. The batch size shows several training samples processed together in a single forward and backward pass through the network (neural network). Smaller batch sizes such as 16 or 32 might lead to some noise and instability during the training procedure. A large batch size like batch size 64 has a more stable and accurate estimation over object detection as it calculates more samples from the dataset. As a result, in the study, we used batch sizes 16 and 64 in some

of the four methods to achieve the best method among the four methods used. We tried to utilize different configurations based on modifying the hyperparameters in each of the four methods. The four methods are represented in the flowchart in Figure 1.

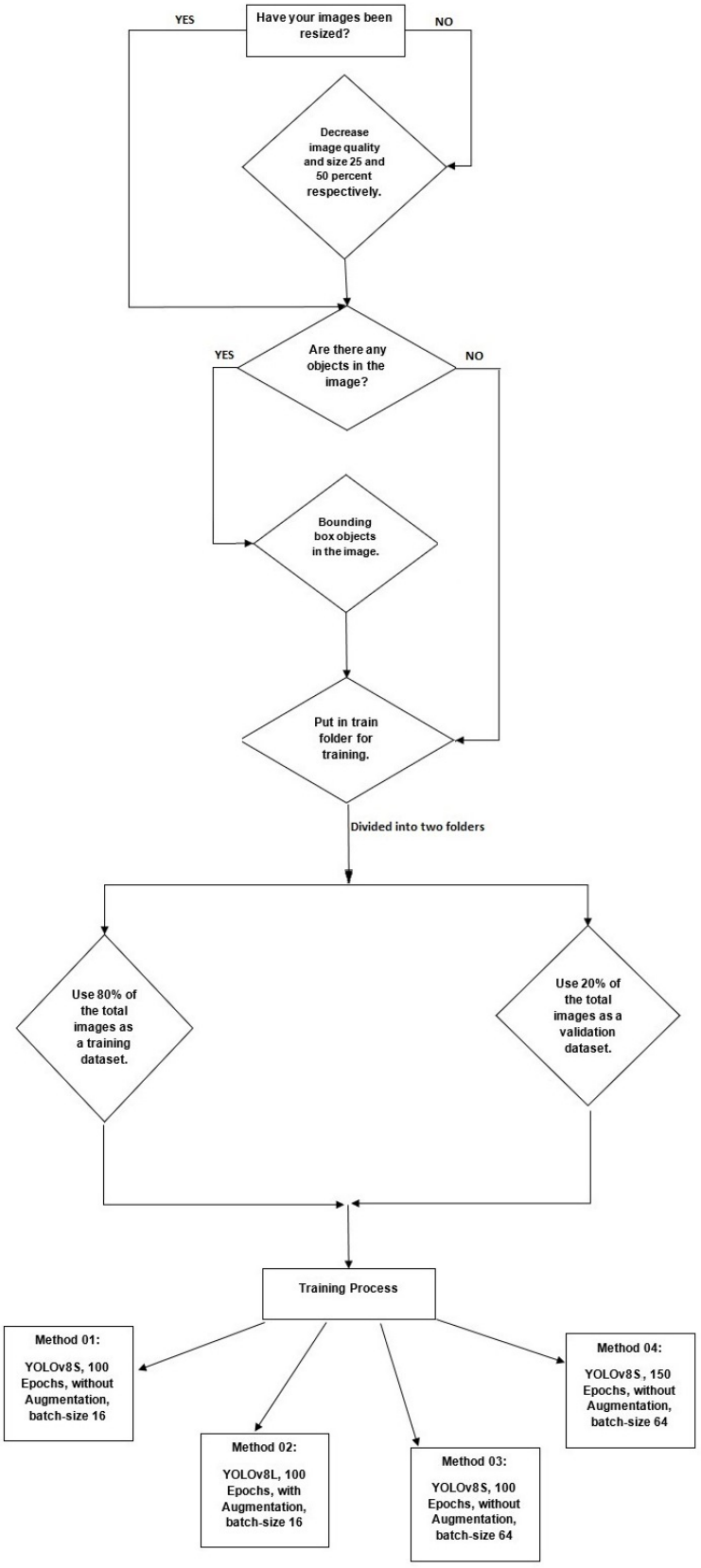

Figure 1. Methodology flowchart.

The four different approaches were used in the proposed methods in Figure 1. First of all, the raw image size and quality were decreased manually. This is not used in common object detection methods. In addition, the proposed approach focuses on challenging conditions in the environment such as foggy weather. We benefited from real foggy images instead of using filters and modules over raw images in order to apply the fog effect in the raw images in the object detection field. Focusing on different weather conditions in the images will have valuable results in object detection projects. Using different weather condition images can be considered as an innovative aspect of our proposed study. The analysis and reporting results are based on various approaches such as using YOLOv8S and YOLOv8L. This can lead to having a comparative performance analysis of different metrics. That is effective and efficient based on time spent and accuracy. The study will have positive effects on object detection in worse weather conditions. It can be used in autonomous vehicles in the future as well. Autonomous vehicles and smart city infrastructure can use the study results where object detection in various weather conditions will be crucial. Other innovative aspects of the proposed study are to improve safety and efficiency and contribute to real-world challenges. This research tries to fill a gap in the object detection field in worse and better weather conditions.

In Figures 2 and 3, there is a noticeable difference in the quality of the traffic light that is placed above the car in the images. In Figure 3, the second image is obtained by Step 1 and exhibits lower resolution. However, the image in Figure 3 is useful to enhance the object detection rate by Yolov8.

Figure 2. One sample of image in foggy weather conditions characterized by 1920 × 1080 pixels stored into training folder.

Figure 3. Image obtained after Step 1 characterized by 940 × 540 pixels.

After the reduction in image quality and size, objects are labeled manually in Step 2. The objects are labeled from 0 to 3 as cars, pedestrians(ped), street signs and traffic lights, respectively. Afterwards, Step 3 is performed and the images in the training dataset are used to start the training step. Conversely, the images moved to the validation dataset (folder) are used for the proposed model validation (not used for unseen images). A sample

of the outcome image moved to the validation dataset (folder) is shown in Figure 4, where the objects are marked in foggy weather conditions.

In the following, we show two images; one is an example outcome from the validation dataset (folder) (see Figure 4) and one is an example of an image predicted by YOLOv8 in training Step 4 (see Figure 5). In Figure 5, the objects are marked by the proposed model with prediction rates in the range 0–1. So, it is clear that the proposed model in the prediction phase works well based on the raw images with a threshold of 0.25 (25% of precision). The objects labeled with a value greater than the 0.25 threshold are detected and marked.

All the required objects are annotated by colorful boxes; vehicles are annotated by red box/es, traffic lights by yellow, pedestrians (crosswalk) by pink and street signs by orange in Figures 4 and 5.

Figure 4. One sample outcome from the validation dataset (folder).

Moreover, Step 4 is performed by a powerful computer on Google Colab and its GPU (Graphics Processing Unit) in which the proposed model is trained with the help of the annotated images obtained by Step 2. In the training step, we use four different methods to improve the object detection results, by considering four configurations of some hyperparameters such as augmentation, batch size and epoch. Moreover, two models of YOLOv8 are used in this methodology and their results are compared with other versions of YOLO in Section 6. YOLOV8S (small) has 11.2 M parameters compared with YOLOv8L (large) which has 43.7 M parameters. YOLOv8L (large) is slower but more accurate. For this reason, we use the small and large model in the proposed detection strategy. In more detail, the following approaches are adopted:

- Method 1: 100 epochs without brightness augmentation, batch size 16 and YOLOv8S (small);
- Method 2: 100 epochs with brightness augmentation, batch size 16 and YOLOv8L (large);
- Method 3: 100 epochs, batch size 64, without brightness augmentation and YOLOv8 (small);

- Method 4: 150 epochs, batch size 64, without brightness augmentation and YOLOv8 (small).

Figure 5. One sample of prediction data.

3.2. Accuracy Analysis

We used features such as true positive, true negative, false positive and false negative to analyze the accuracy and performance of the proposed trained model. A positive is when the required objects are identified and detected in images with the help of a bounding box. A negative is when there is no detection in the images. True and false explain the detection correctness of the desired objects, which must be recognized in the images. True negative occurs when there are no desired objects in the images and hence they are not realized; false positive occurs when objects are discovered by mistake as desired objects. When a model fails to recognize desired items, this is referred to as a false negative. The information allows us to determine how well the dataset has been trained [27,28,30]. We consider the Precision , Recall and Precision–Recall metrics to analyze the accuracy of the proposed model:

$$Precision = \frac{(True\ Positive)}{(True\ Positive\ +\ False\ Positive)} \tag{1}$$

$$Recall = \frac{(True\ Positive)}{(True\ Positive\ +\ False\ Negative)} \tag{2}$$

$$Accuracy = \frac{(True\ Positive\ +\ True\ Negative)}{(True\ Positive\ +\ True\ Negative\ +\ False\ Positive\ +\ False\ Negative)} \tag{3}$$

Precision means calculating the number of positive samples correctly detected to a total number of samples as positive either correctly or incorrectly detected. A proposed model will work very well and achieve high accuracy if it shows many correct positives or a few false positives. Recall is the calculated rate between the number of positive samples correctly detected and the total number of positive and negative samples. The recall measures the model's ability to detect positive samples [12,21]. Loss means a proposed model's bad prediction about detecting the objects or classes and if it decreases continually near zero, we will have a good prediction model [5]. As well as the precision, recall and precision–recall metrics that are used to check an object detection model, we can use mAP50

and mAP50-95 metrics too. The mAP50 (mean Average Precision at 50%) measures the average precision of detection across different classes when the intersection over union (IoU) threshold is set to 0.5. The mAP50-95 (means Average Precision from 50 percent to 95 percent) calculates the average precision over a range of IoU thresholds from 0.5 to 0.95 with a step size of 0.05. This metric gives a more comprehensive evaluation of the model performance across a range of IoU thresholds. It is particularly useful for understanding how well the model performs at different levels of localization accuracy. In this paper, we used the precision, recall, precision–recall and mAP50 metrics for model accuracy evaluation.

When dealing with images, there can be unwanted elements, called noise. In DL, we adjust some settings, such as how the data are modified or how many times the model learns (epochs), to make our model more accurate and reduce errors when checking against new data. For example, we can change images in various ways, like rotating them or adjusting brightness, to make our model better at recognizing objects [5,12,25]. In our model, we used brightness augmentation, which increased the number of images to about double. This helps YOLOv8 to spot objects in images. Making these changes can also prevent spending more time or less time on the learning step from the data. Moreover, we re-size all the images to make spotting objects easier. Then, we split the images into two parts (folders): 80 percent is used to train the model, and the remained 20 percent is used to check how good the model is at detecting the objects [30].

Batch size is the number of samples processed before a proposed model is updated, and augmentation is a method of increasing a database using tactics such as reverse, crop photos, raise brightness and so on in images [17]. A suggested model may be trained if we have a clear image and adequate brightness, and objects in photographs on streets can be spotted. Brightness and sufficient light in images are two other crucial keys in ML and DL [8,9,12]. Object size is also a crucial measure for training a proposed model [9], and object identification accuracy will have a favorable effect on road traffic management systems [8].

4. Case Study

In order to test the proposed methodology, we use a 9 gigabyte dataset of 3.806 images recorded by a dashboard camera in foggy weather conditions in Zurich: the training dataset includes 3045 images, while the validation dataset is composed of 761 images equal to 20% of the total. The image resolution is 1920 $\times$ 1080 pixels and the average size is about 2 megabytes.

After the first three steps, we perform the training step and evaluate each proposed method considering precision, recall, precision–recall and final result items.

In method 1, we use 100 epochs, batch size 16, without augmentation technique and YOLOv8S (small). The data training process takes 4 h. Overall, in method 1, the precision–confidence metric is 94.5% for all classes as shown in Figure 6. The precision metric vs. epoch is represented in Figure 7, where the precision metric visibly improves from a minimum point of 65% in the third epoch to a pick point of about 86%. From epoch 1 to epoch 20, the precision metric increases slightly from 78% to 82%. Between epochs 20 and 80, the metric goes up from 82% to 84%. Afterward, the precision metric fluctuates at around 86%.

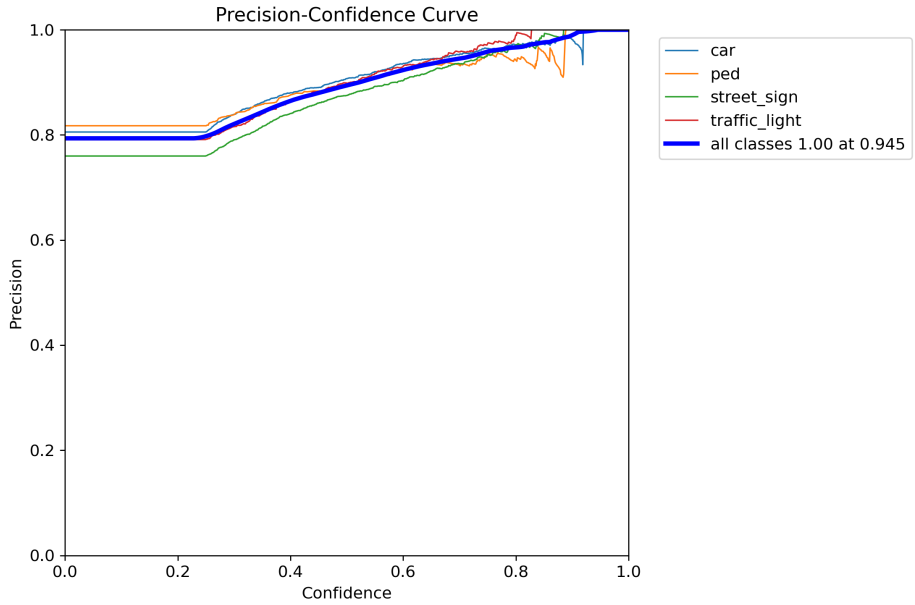

Figure 6. Precision–confidence curve for method 1 (100 epochs, batch size 16, without augmentation and YOLOv8S).

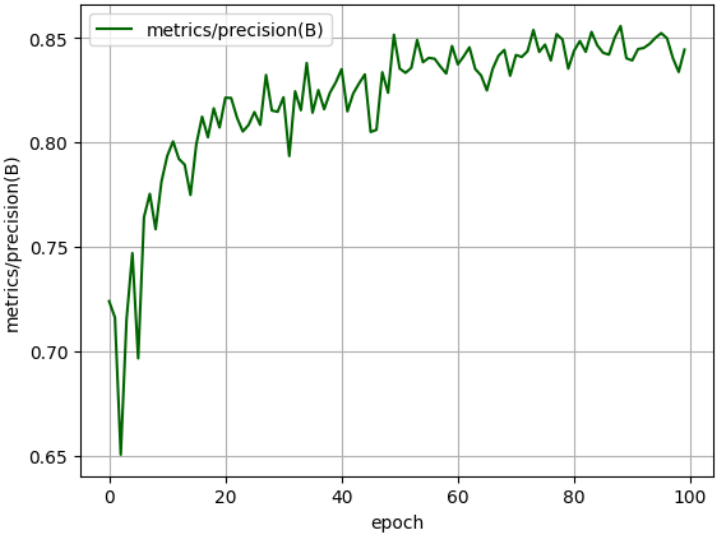

Figure 7. Precision vs. epoch for method 1.

Recall–confidence, as a second accuracy metric, improves during method 1 and is about 81% for all classes, and it is presented in Figure 8. In Figure 9, the recall metric vs. epochs is reported. At the beginning, in epoch 3, the recall feature is at about 50%. From epoch 5 to 60, the recall significantly grows from 50% to nearly 76%. Between epochs 60 and 100, there is a slight increase in the recall feature from about 76% to about 78% as shown.

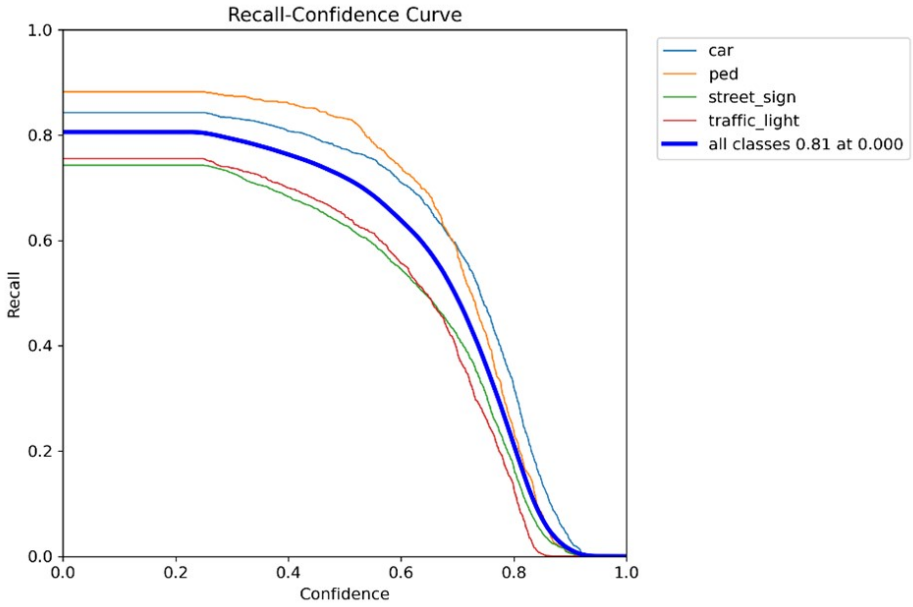

Figure 8. Recall–confidence curve for method 1 (100 epochs, batch size 16, without augmentation and YOLOv8S).

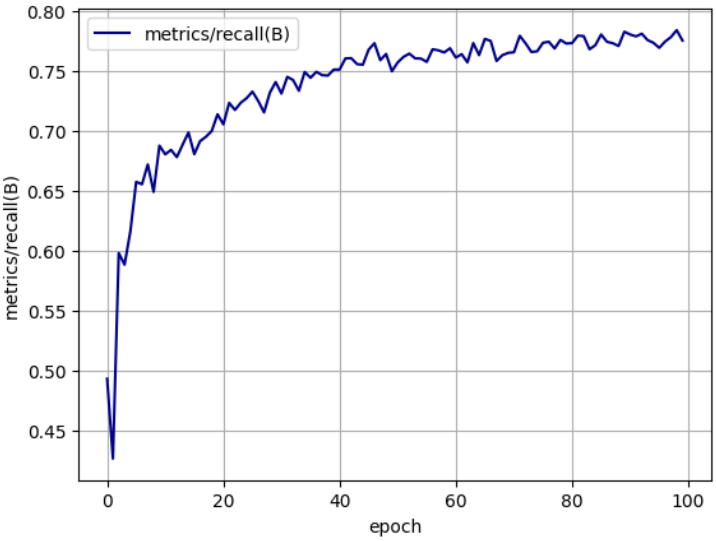

Figure 9. Recall vs. epoch for method 1.

Figure 10 reports the third metric, i.e., the precision and recall curve: cars, pedestrians, street signs and traffic lights are 87.0%, 87.9%, 78.8% and 81.2%, respectively. It was 83.7% on average for all classes.

Finally, Figure 11 shows the precision, recall and mAP50 metrics for method 1 whereas all loss metrics decrease significantly.

The method 1 outcome is shown in Table 1. The following abbreviations are used in the future: EP = number of epochs, BS = batch size, Aug = augmentation and YOLOv8s = YOLOv8 (small).

Table 1. Prediction rate for detecting objects in method 1.

Method	Car %	Pedestrian %	Street Sign %	Traffic Light %
100 EP, BS 16, NO AUG, YOLOv8S	87.0	87.9	78.8	81.2

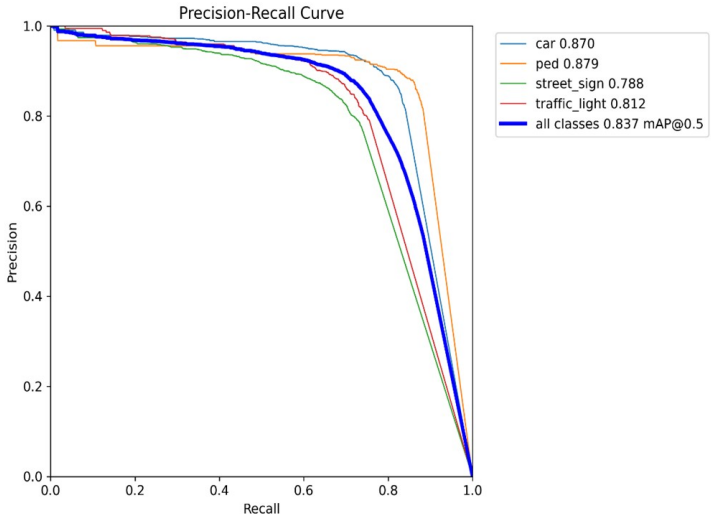

Figure 10. Precision–recall curve for method 1 (100 epochs, batch size 16, without augmentation and YOLOv8S).

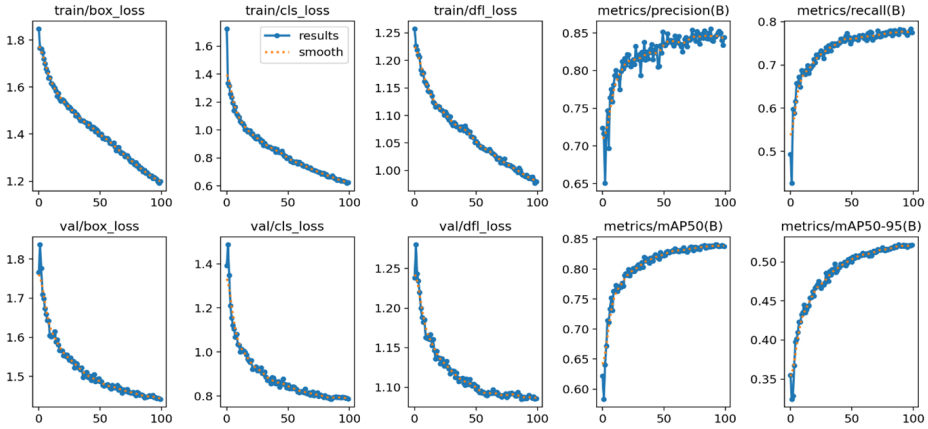

Figure 11. Final result for method 1 (100 epochs, batch size 16, without augmentation and YOLOv8S).

In method 2, we use 100 epochs and batch size 16 as in method 1, but we utilize the augmentation technique in YOLOv8L (large), whereas YOLOv8S (small) is used in method 1. YOLOv8L has more layers than YOLOv8S. The precision metric result is 94.5%, equal to the same metric in method 1, and it is presented in Figure 12. In precision vs. epoch, in epoch 3, the precision metric is placed at a minimum level of about 74%. From epoch 3 to 20, the trend increases to about 83%. Between epochs 20 and 60, the graph enhances slightly from 84% to 85%. Afterward, the precision metric fluctuates at around 86%. In Figure 13, we report the trend of precision with respect to epochs that are 87 and not 100, because Google Colab's running time limitation leads to the loss of 13 epochs. Overall, the recall metric, as a second accuracy metric, increases to 83% for all classes and it can be seen in Figure 14. The recall metric improves by 2% compared to method 1. In Figure 15, the recall vs. epoch trend is shown. In epoch 3, the recall metric is about 59% which is at its minimum level; from epoch 3 until 40, the metric advances substantially from 59% to nearly 79%. Between epochs 40 and 87, the trend fluctuates at around 83%. The precision–recall metric as a third accuracy metric is increased in the second approach. The car detection rate increases by 1.4% and reaches 88.4%. Pedestrian detection goes up slightly by 1.1% and reaches 89%. The street signs detection ratio increases by 1.9%. The ratio in traffic lights climbs to 85.1% whereas the metric in method 1 is 83.7%. Finally, overall, the object detection ratio for all classes increases to 85.1% compared with the same metric in the first technique which is 83.7% and it is presented in Figure 16. This indicates that

YOLOv8L has more layers compared with YOLOv8S. The results improve in the second approach compared to method 1. The large version of the YOLO model has positive effects on the object detection results. Finally, the final result of the second approach is shown in Figure 17. Unfortunately, method 2 faces an overfitting/underfitting problem in the validation distribution focal loss metric (val/dfl-loss) results.

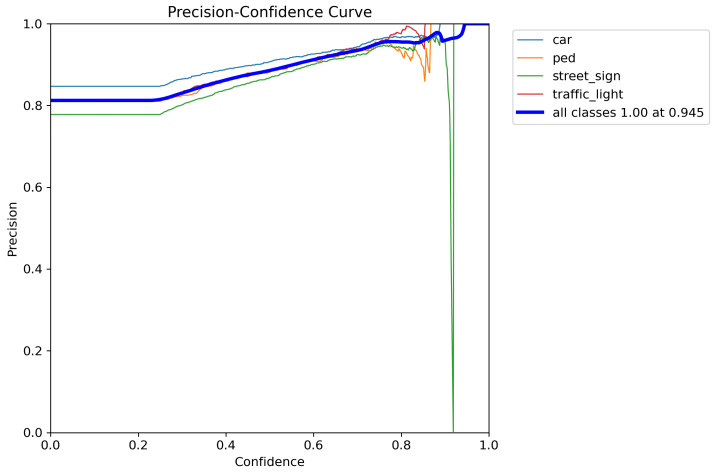

Figure 12. Precision–confidence curve for method 2 (100 epochs, batch size 16 and with augmentation, YOLOv8L).

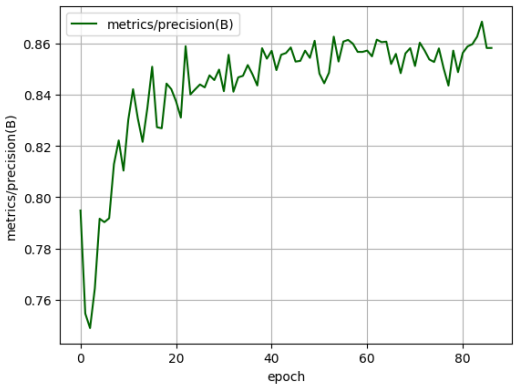

Figure 13. Precision vs. epoch for method 2.

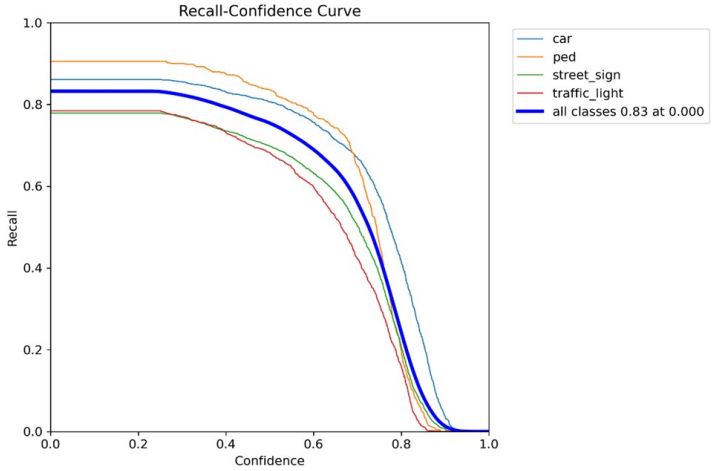

Figure 14. Recall–confidence curve for method 2 (100 epochs, batch size 16 and with augmentation, YOLOv8L).

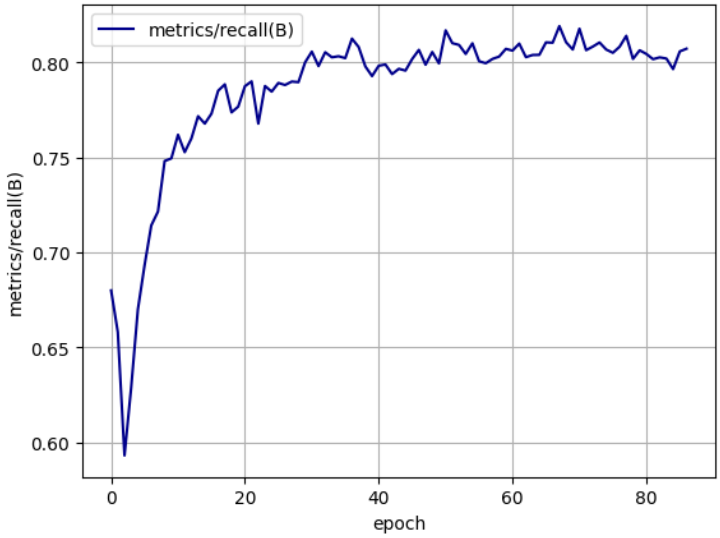

Figure 15. Recall vs. epoch for method 2.

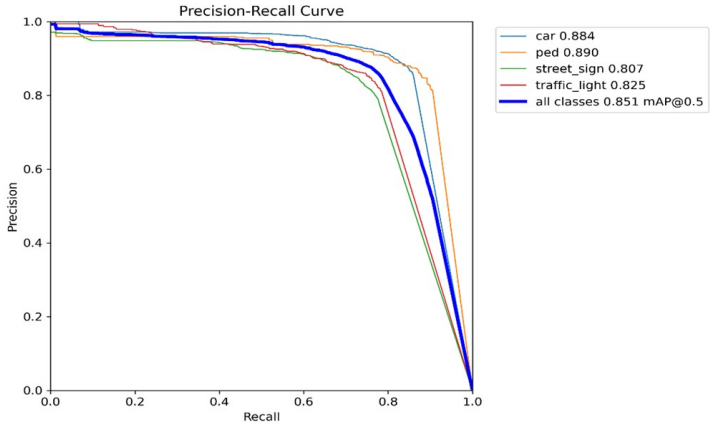

Figure 16. Precision–recall curve for method 2 (100 epochs, batch size 16, with augmentation and YOLOv8L).

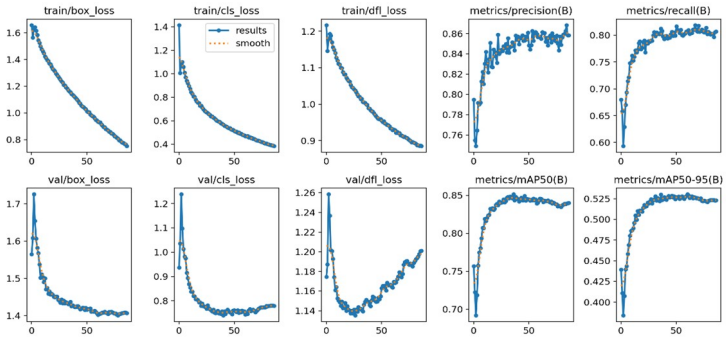

Figure 17. Result for method 2 (100 epochs, batch size 16 and with augmentation, YOLOv8L).

The method 2 outcome is shown in Table 2.

Table 2. Prediction rate for detecting objects in method 2.

Method	Car %	Pedestrian %	Street Sign %	Traffic Light %
100 EP, BS 16, AUG, YOLOv8L	88.4	89.0	80.7	82.5

In method 3, we use 100 epochs, batch size 64, without augmentation and YOLOv8S (small). It take about 2.5 h for the model to be trained. Method 3 benefits from batch size 64 without using augmentation technique rather than batch size 16. The precision–confidence metric in the third technique is lower than the metric in the two previous strategies and it is equal to 93.2% as presented by Figure 18. The metric starts from epoch 3; its value is about 57%. From epoch 3 to 40, the precision increases by about 20% and it reaches about 81%. Between epochs 40 and 80, the metric grows gradually to approximately 84%. Then, between epochs 80 and 100, it fluctuates at around 84%. This is shown in Figure 19.

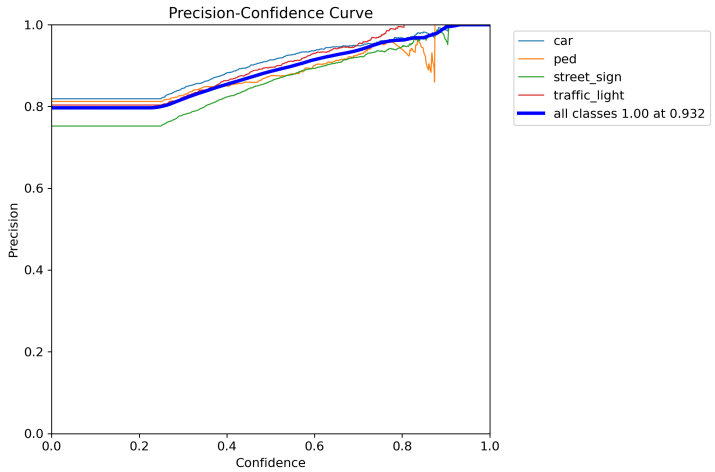

Figure 18. Precision–confidence curve for method 3 (100 epochs, batch size 64 and without augmentation, YOLOv8S).

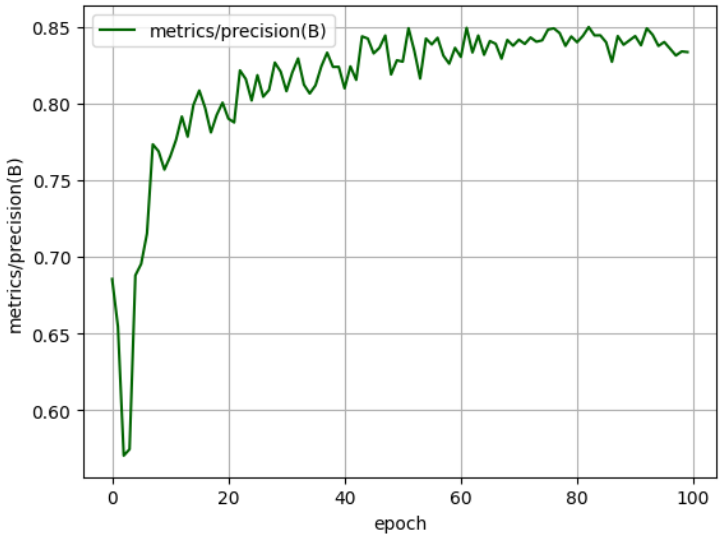

Figure 19. Precision vs. epoch for method 3.

The recall–confidence metric, as a second accuracy feature, was 81% for all classes and it is shown in Figure 20. Its trend shows the method is able to detect objects well. In Figure 21, the recall vs. epoch trend can be seen. The recall metric in epoch 3 is about 32%. From epoch 3 to epoch 40, it increases rapidly to about 76%. Between epochs 40 and 60, it improves marginally by 1% and reaches 77%. Afterward, it fluctuates to around 78%. In the precision–recall metric, as a third accuracy metric, car, pedestrian, street sign and traffic light detection rates are 86.8%, 88.9%, 78.3% and 81.8%, respectively. The class rate in the precision–recall metric is 84.0% and it can be seen in Figure 22. Method 3 improves the precision–recall metric considering pedestrians, traffic lights, and overall detection rates compared to method 1. The final results for method 3 are presented in Figure 23.

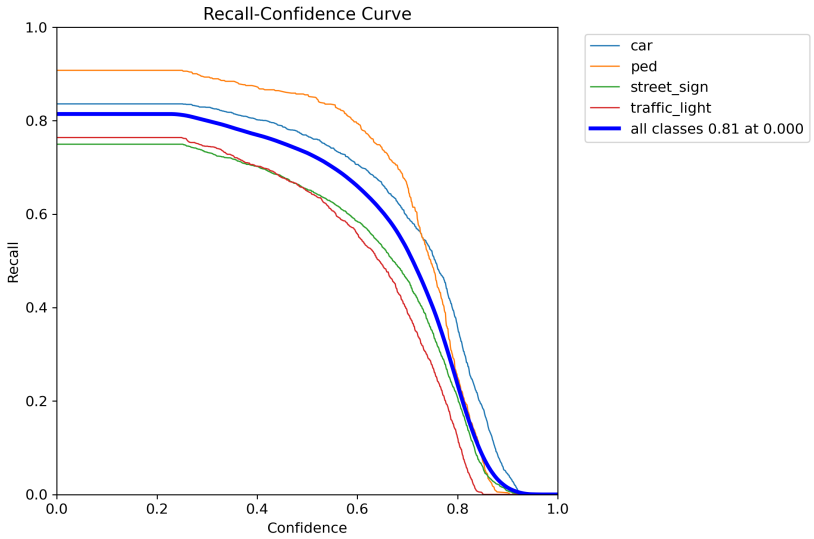

Figure 20. Recall–confidence curve for method 3 (100 epochs, batch size 64 and without augmentation, YOLOv8S).

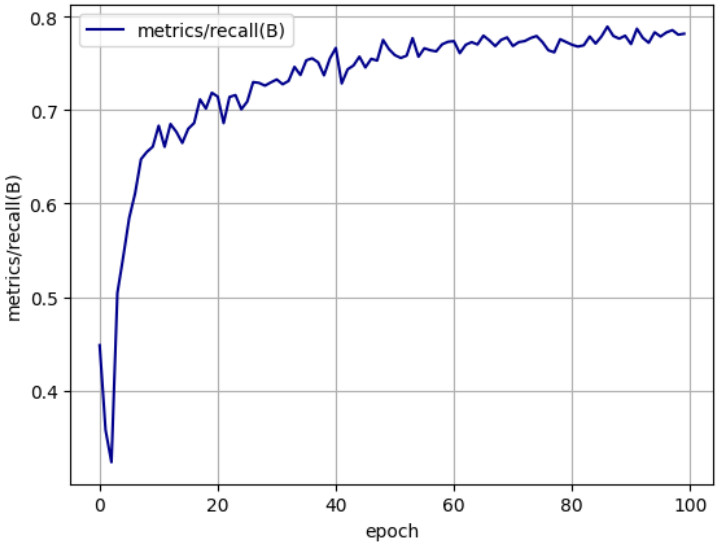

Figure 21. Recall vs. epoch for method 3.

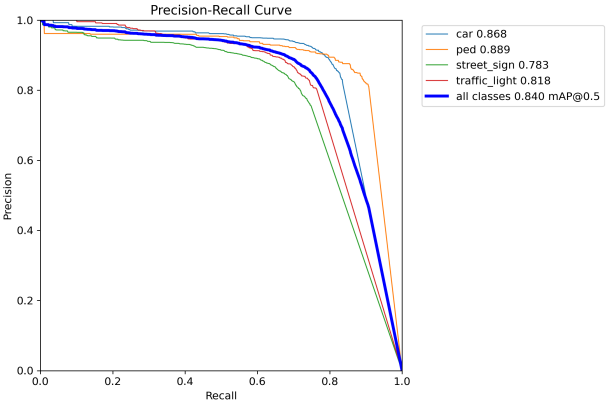

Figure 22. Precision–recall curve for method 3 (100 epochs, batch size 64 and without augmentation, YOLOv8S).

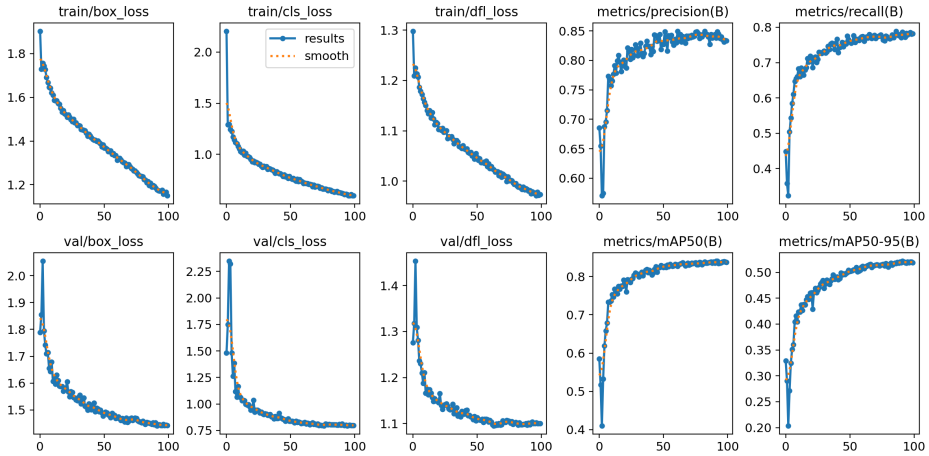

Figure 23. Result for method 3 (100 epochs, batch size 64 and without augmentation, YOLOv8S).

The method 3 outcome is shown in Table 3.

Table 3. Prediction rate for detecting objects in method 3.

Method	Car %	Pedestrian %	Street Sign %	Traffic Light %
100 EP, BS 16, AUG, YOLOv8S	86.8	88.9	78.3	81.8

In method 4, we benefit from 150 epochs, batch size 64 and without augmentation. It takes 3 h for the model to be trained. In method 4, most hyperparameters increase, such as YOLOV8L (more layers), batch size and epochs. Increasing these hyperparameters has positive effects on the final results. As a first accuracy metric, the precision–confidence metric is improved to 95.1%. This is the highest value among all the approaches and it is shown in Figure 24. In the precision vs. epoch image, in epoch 3, the precision metric is about 60%. From epoch 3 to epoch 20, the metric increases by about 20%. Between epochs 20 and 80, it enhances slightly from 80% to nearly 85%. Afterward, it fluctuates at around 85% and it is shown in Figure 25. The recall metric, as a second accuracy metric for the fourth plan, for all classes is 82% as shown in Figure 26. In Figure 27, the recall vs. epoch trend is shown. The recall, in epoch 2, is about 30%. From epochs 2 to 20, there is a significant increase to nearly 70%. From epoch 20 until 80, the metric climbs to about 77%. Between epochs 80 and 150, the recall metric levels off at approximately 78%. For the precision–recall item, as a third accuracy metric, the results for cars, pedestrians, street signs, traffic lights and overall class rates are 87.7% , 89.4% , 78.7% , 82.0% and 84.4%, respectively, and are shown in Figure 28. The precision–recall metric trend is the confirmation that the proposed method works well in detecting objects from the images. The outcomes of the fourth scenario are shown in Figure 29. The precision–recall metric has a better performance compared with the first method and method 3. By contrast, it has worse results compared with the second technique.

The method 4 outcome is shown in Table 4.

Table 4. Prediction rate for detecting objects in the fourth method.

Method	Car %	Pedestrian %	Street Sign %	Traffic Light %
150 EP, BS 64, No AUG, YOLOv8S	87.7	89.4	78.7	82.0

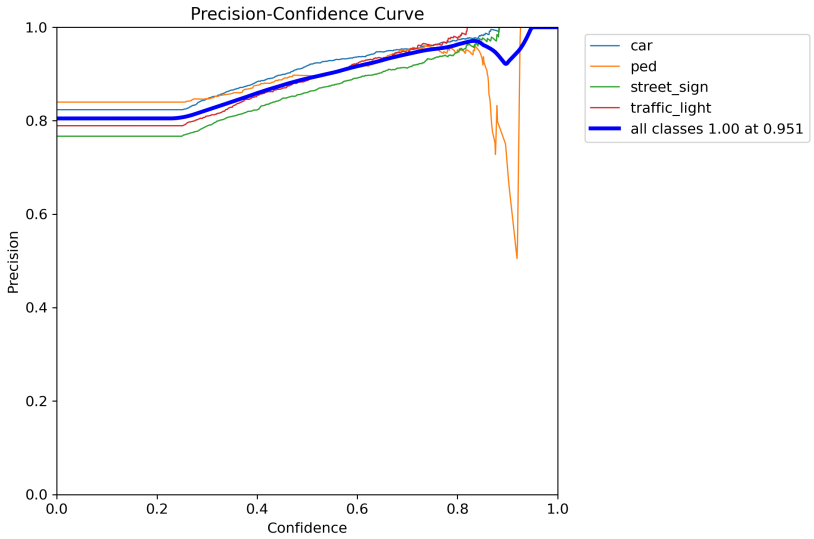

Figure 24. Precision–confidence curve for method 4 (150 epochs, batch size 64 and without augmentation, YOLOv8S).

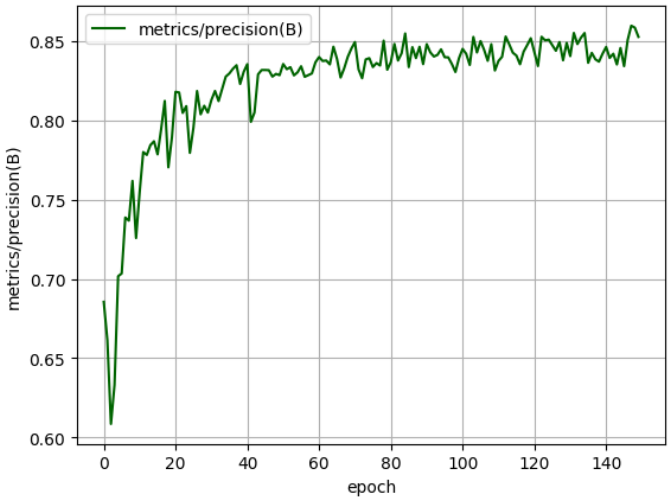

Figure 25. Precision vs. epoch for method 4.

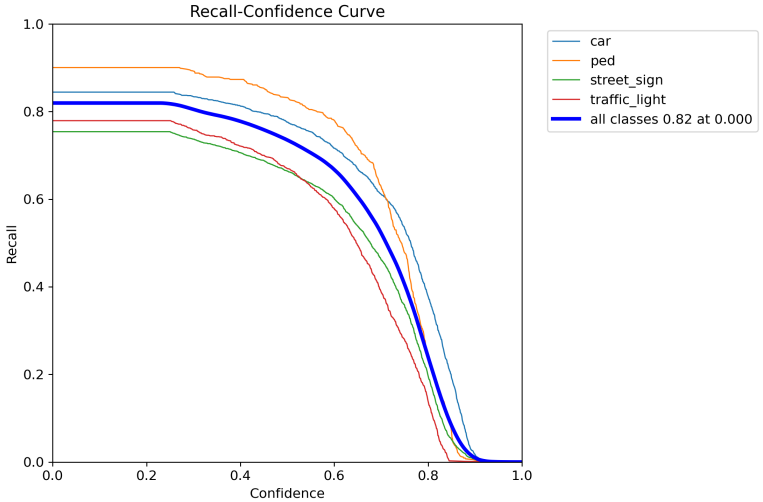

Figure 26. Recall–confidence curve for method 4 (150 epochs, batch size 64 and without augmentation, YOLOv8S).

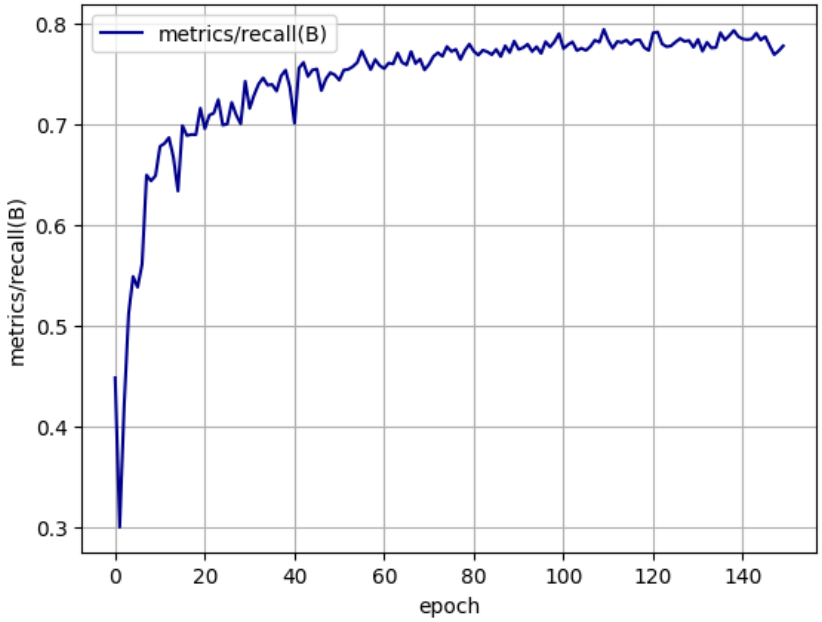

Figure 27. Recall vs. epoch for method 4.

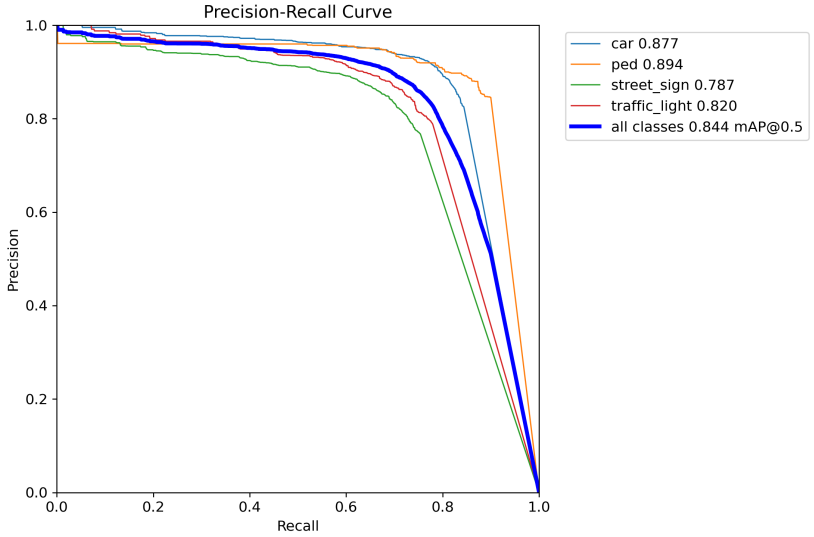

Figure 28. Precision–recall curve for method 4: (150 epochs, batch size 64 and without augmentation, YOLOv8L).

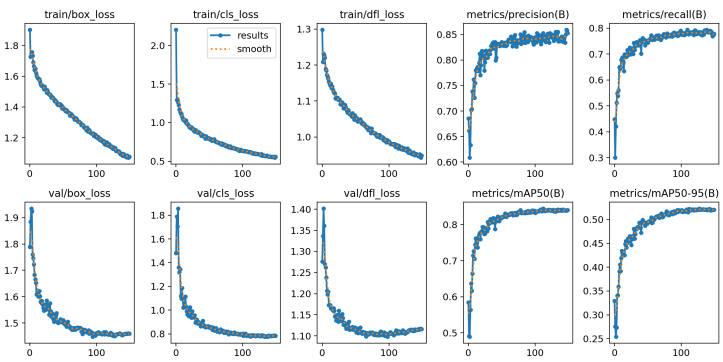

Figure 29. Result for method 4 (150 epochs, batch size 64 and without augmentation, YOLOv8S).

According to the analysis of all the results, the four different techniques detect objects in the images. Method 2 has the maximum precision, recall and mAP50 metrics among all methods. Method 4 has the second rank among all approaches and it has better results in precision–recall and overall detection rate for all classes compared with the first method and method 3 for all objects.

5. Discussion

The precision, recall and mAP50 features were analyzed for the four proposed methods and are summarized in Table 5. Note that NO AUG means that, in that method, the augmentation method was not used and AUG means that the method benefited from the augmentation technique. As demonstrated by the results in Table 5, augmentation in method 2 led to access to higher object detection ratios in the precision, recall and mAP50 metrics. If the number of epochs is increased in the training process, the metric percentages improve. In the fourth approach, 150 epochs were employed, and the method outcome ratio was higher than that in the first method and method 3. In addition, the required object detection accuracy percentages are in Tables 1–4 for methods 1 to 4, respectively.

Table 5. Precision, recall and mAP50 metrics for the four proposed methods in the foggy dataset.

Methods	Precision %	Recall %	mAP50 % (Average for All Desired Objects)
100 EP, BS 16, NO AUG, YOLOv8S	82.0	73.7	83.7
100 EP, BS 16, AUG, YOLOv8L	84.3	78.5	83.9
100 EP, BS 64, No AUG, YOLOv8S	81.2	72.95	83.7
150 EP, BS 64, No AUG, YOLOv8S	82.3	74.3	84.0

The object detection methods used in other articles are illustrated in Table 6. In the other articles, all classes of objects including cars, pedestrians, traffic lights, and traffic signs have not been detected at the same time. Some abbreviations are used in the columns of Table 6 including : S for sunny weather, R for rainy weather, F for foggy weather, YL equal to YOLO, FRN for Faster-RCNN (Region-based Convolutional Neural Network), and RN for RCNN (Region-based Convolutional Neural Network). Some parts of the Table 6 are marked with a dash, since the data were not available or were not mentioned in the articles. As can be seen, in most of the articles, the images used were in sunny or good weather conditions. They used more datasets or images in their studies rather than our proposed model which uses just one dataset that includes about 4000 foggy road images.

By considering Tables 5 and 6, we will be able to have a comparison between the four proposed methods and other methods used in other articles by other scientists. As can be seen, in the majority of articles, the authors used more epochs than in our research. Also, the number of images used in other studies is higher than ours. In addition, if we had more images, it is more likely we would have better outcomes in research regarding the object detection field.

To prove the proposed model robustness, we tested various images from three other datasets. The datasets included sunny and good weather road images. We concatenated and tested the three datasets in a new, unique dataset as the test dataset. The front and rear images of car, Semantic Segmentation Makassar (IDN) Road and traffic sign datasets in YOLO format datasets were downloaded from the Kaggle website and were utilized to test the proposed model. The new dataset included a total of about 430 images. In the following lines, some images will be added to show the proposed model accuracy in identifying the objects in the sunny and good weather images as well. The outcomes will illustrate the ability of the proposed model to detect objects. The test images were added. Cars are red-colored bounding boxes, street signs are orange-colored, traffic lights are yellow-colored and pedestrians (crosswalks) are blue-colored by the proposed model. In Figure 30, as can be seen, cars and traffic signs were identified by the proposed model with an accuracy of 92 %,63%, 88%, 66% and 87%, respectively, in sunny weather images on the road.

Table 6. Object detection methods and classes used in other articles.

Methods	Number of Epochs	Precision% (Average Precision Methods or Specific Objects)	Recall % (Average Recall Methods or Specific Objects)	mAP50 % (Average mAP50 of the Methods or Specific Objects)	Total Number of Images	Class Names
[5], YLv3, S, R	120	47.4	33.7	44.47	20,000	Multi-objects
[9], YLv7-tiny, S	100	-	-	80.02	16,441	Bike, motorcycle, bus, car and people
[17], YLv2, 3, FRCNN, S	-	-	-	48.6	13,427	Traffic light status (green, yellow, red and off)
[11], YLv3, 4, FRN, RN, S	-	70.14	85.01	69.01	2,000,000	Car, truck, van, tricycle, bike, pedestrian, people, bus, motor, awning-tricycle
[18], YLv5, S	80	-	-	43.3	5000	Person, rider, car, truck, bus, train, motor, bike
[7], YLv3, S	120	63.0	55.0	46.60	330,000	Car, truck, pedestrian, traffic light and traffic sign
[1], YLv8s, Bi-PAN-FPN and GhostblockV2, S	150	79.5	79.2	83.7	8629	People, pedestrian, car, van, truck, tricycle, bus, motor, bike, awning-tricycle
[19], YLv5, 7, 8 and YLR, S	250	79.37	44.16	36.49	8000	Bus, bike, car, pedestrian, truck
[23], Faster-RCNN, S	120,000	-	-	-	383,240	People
[20], FRN+ Online Hard Examples Mining (OHEM), S	70,000	-	-	44.46	5000	Traffic sign
[25], FRN, YLv5, DETR, S	200	98.94	72.22	-	7500	Multi-objects
[26], YLv5, 6, 7, YLGW, S, R, F	300	-	87.84	-	100,000	People
[10], FRN, S	150,000	-	87.7	80.29	110,000	Traffic light
[21], YLv3, S	150,000	71.51	69.05	63.80	95,000	Traffic light status (none, off, red and green)

Figure 30. Detected objects by the proposed model in Semantic Segmentation Makassar (IDN) Road Dataset—the second test dataset.

Figure 31 shows the required objects selected in the picture, including cars, traffic lights and traffic signs. There is a lack of object detection in the image. However, it must be considered that the objects were detected with the help of a YOLOv8 model that was trained for foggy images, not sunny images. It is difficult, for some objects, such as cars, pedestrians, traffic signs and traffic lights, to be marked by this model. Sunny image patterns and object color contrast are different from the patterns and object color contrast in foggy images.

Figure 31. Detected objects from traffic signs dataset in YOLO format—the third test dataset.

Also, Figure 32 was captured in the suburbs, a different environment from our image dataset. The various environmental images can pose a new challenge for the proposed model. Also, it demonstrates the robustness of the proposed object detection model.

The cars were detected by the proposed model with an accuracy of 88%. But, unfortunately, another car and some traffic signs have not been detected on the suburb road.

Figure 32. Detected car in suburbs from traffic signs dataset in YOLO format.

Figure 33 has a complicated background including buildings and trees: the proposed model was able to identify the objects correctly.

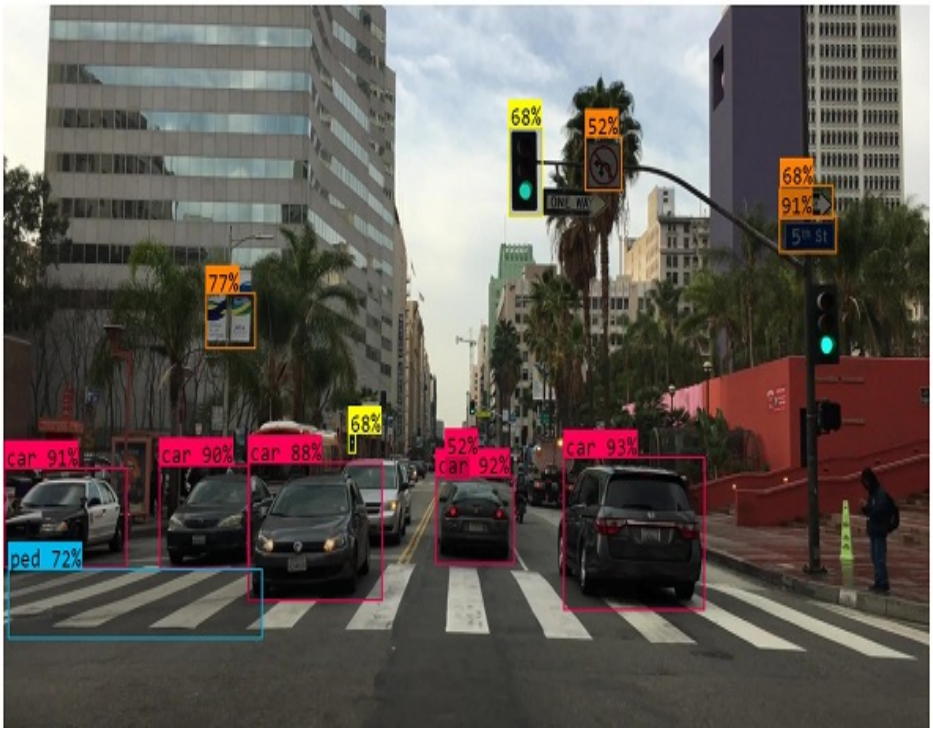

Figure 33. Detected objects from front and rear image of car dataset with complex background—the first test dataset.

Table 7 summarizes the accuracy of the detected objects correctly in the tested datasets. Note that zero values mean that objects are not present or the object was not detected by the model.

Table 7. The predicted object detection accuracy rate by the proposed model in good and sunny images in the test dataset (good and sunny image datasets).

Dataset Name	Car (%)	Pedestrian (%)	Street Sign (%)	Traffic Light (%)
1-Front and rear image of car	80.74	0.00	70.95	0.00
2-Semantic Segmentation Makassar(IDN) Road Dataset	77.09	67.50	71.23	67.55
3-Traffic signs dataset in YOLO format	73.86	0.00	72.34	68.42
Overall detection rate in the three datasets	77.23	67.50	71.35	67.98

In addition, we trained the new test dataset (430 sunny and good images) based on the prior structures, and hyperparameters were used in the foggy image dataset (including the method with 1100 epochs, batch size 16, no augmentation and YOLOv8S; the method with 2100 epochs, batch size 16, with augmentation and YOLOv8L; the method with 3100 epochs, batch size 64, no augmentation and YOLOv8S; and the method with 4150 epochs, batch size 64, no augmentation and YOLOv8S). The outcomes will be illustrated in the following figures: Figures 34–37 for the four methods used in the foggy image dataset. As can be seen, the outcomes of method 1 (100 epochs, YOLOv8S, batch size 16 and no augmentation) will be presented in precision, recall, precision–recall and final report images, respectively.

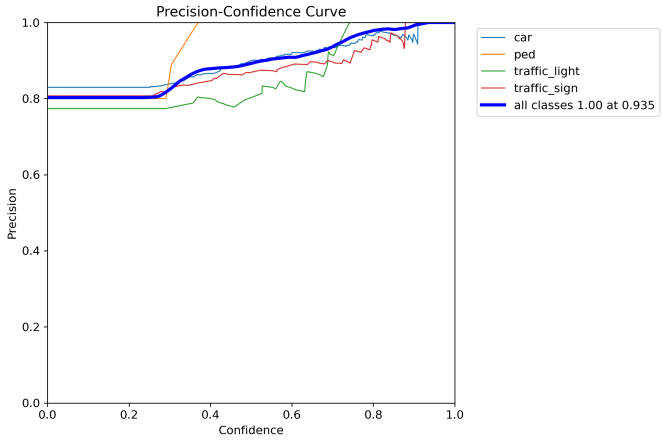

Figure 34. Precision rate for method 1 in test dataset (good and sunny dataset).

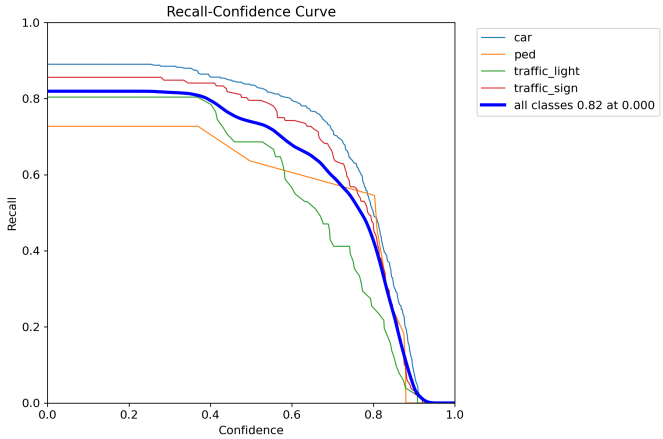

Figure 35. Recall rate in method 1 in the test dataset (good and sunny dataset).

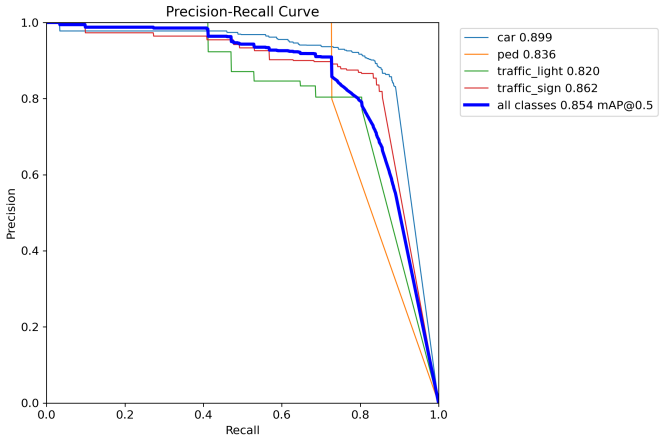

Figure 36. Precision–recall rate for method 1 in test dataset (good and sunny dataset).

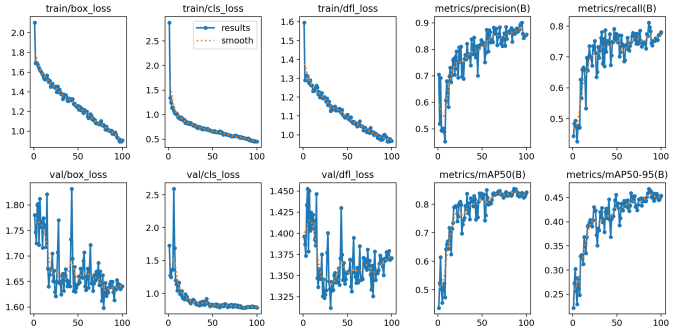

Figure 37. Final report rate in method 1 in the test dataset (good and sunny dataset).

The second approach was also tested in the test dataset. Method 2 includes 100 epochs (100EPC), batch size 16 (BS16), augmentation (AUG) and YOLOv8L. In the following figures, Figures 38–41, precision, recall, precision–recall and the final report, respectively, will be displayed in the images.

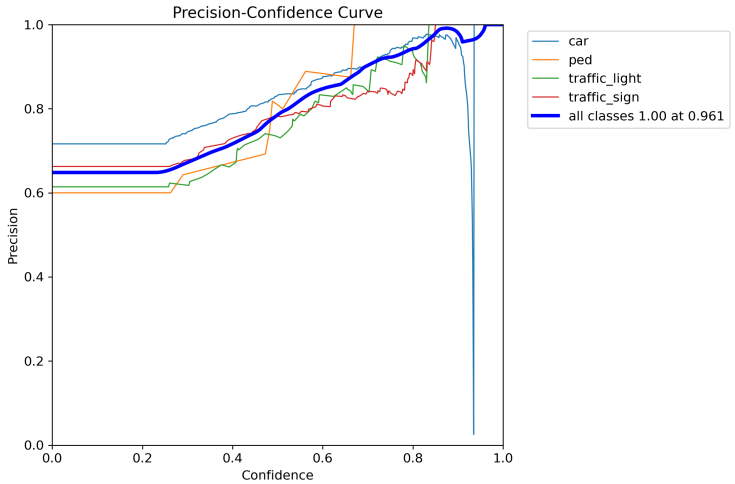

Figure 38. Precision rate in the second approach in sunny and good image dataset (100 EPC, BS16, AUG and YOLOv8L in test dataset).

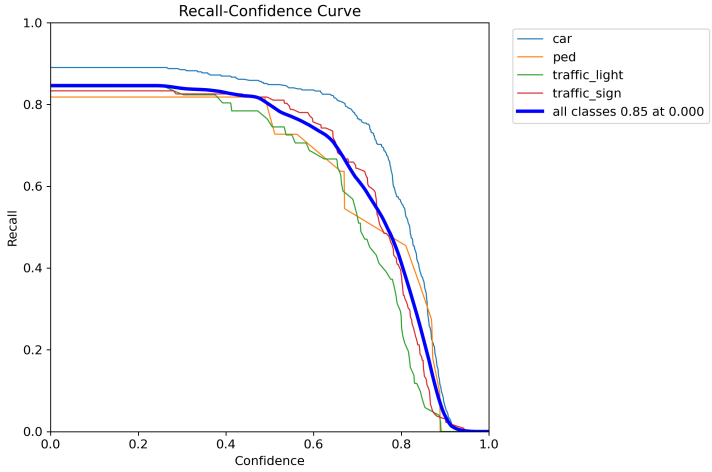

Figure 39. Recall rate in the second approach in sunny and good image dataset (100 EPC, BS16, AUG and YOLOv8L in test dataset).

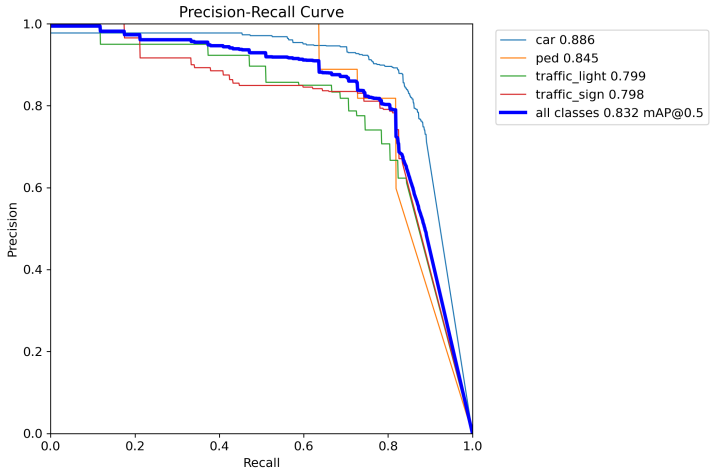

Figure 40. Precision–recall rate in the second approach in sunny and good image dataset (100 EPC, BS16, AUG and YOLOv8L in test dataset).

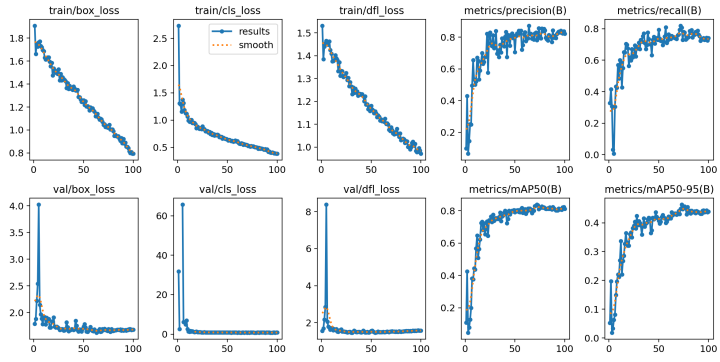

Figure 41. The final report in the second approach in sunny and good image dataset (100 EPC, BS16, AUG and YOLOv8L in test dataset).

The third technique was applied to test the test dataset (a sunny and good image dataset) with the help of 100 epochs, batch size 64, no augmentation and YOLOv8S. The results are shown in Figures 42–45 for precision, recall, precision–recall and the final report, respectively.

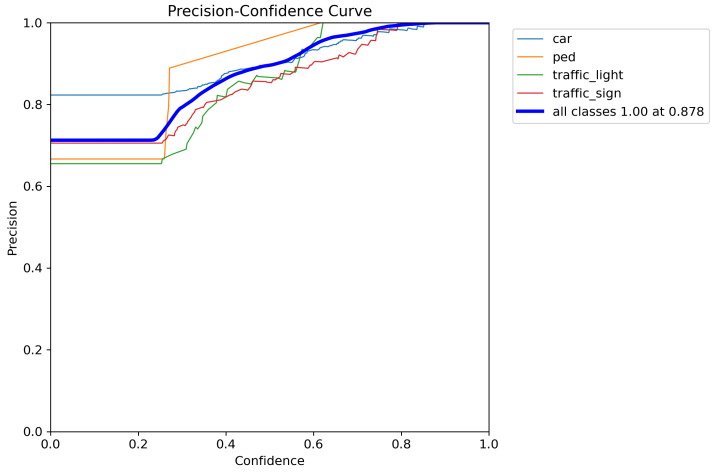

Figure 42. Precision rate in the third approach in sunny and good image dataset (100 EPC, BS16, No AUG and YOLOv8S in test dataset).

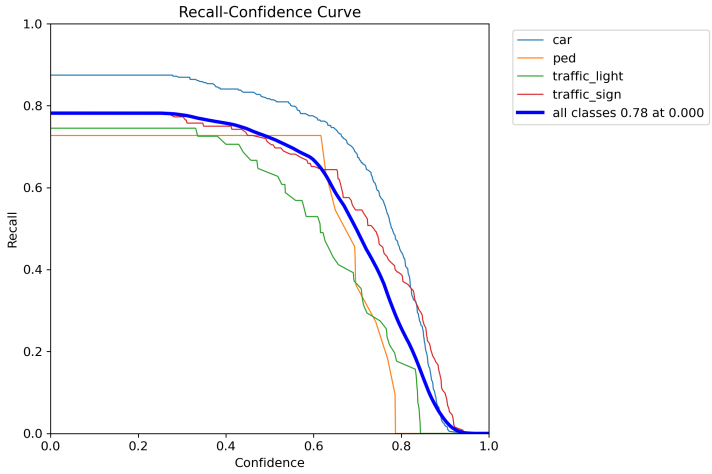

Figure 43. Recall rate in the third approach in sunny and good image dataset (100 EPC, BS16, No AUG and YOLOv8S in test dataset).

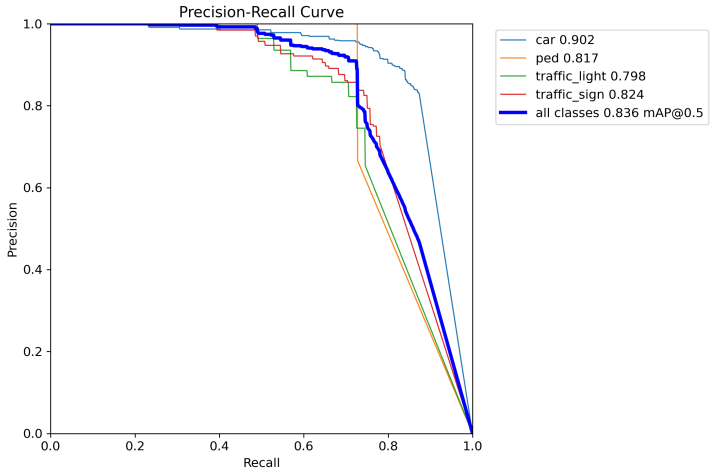

Figure 44. Precision–recall rate in the third approach in sunny and good image dataset (100 EPC, BS16, No AUG and YOLOv8S in test dataset).

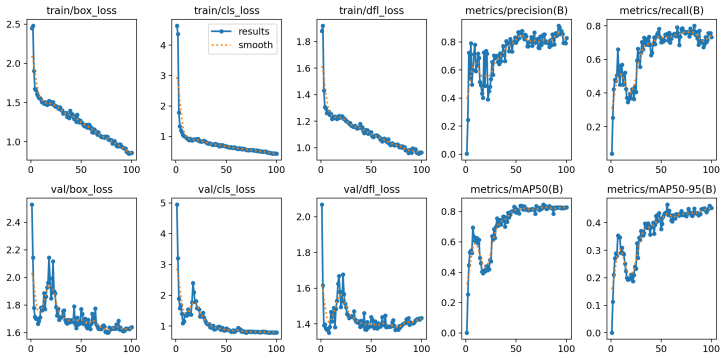

Figure 45. The final report in the third approach in sunny and good image dataset (100 EPC, BS16, No AUG and YOLOv8S in test dataset).

The final approach was utilized over the test dataset (a sunny and good image dataset) consisting of 150 epochs, batch size 64, no augmentation and YOLOv8S. The outcomes are shown in Figures 46–49 as precision, recall, precision–recall and the final report individually.

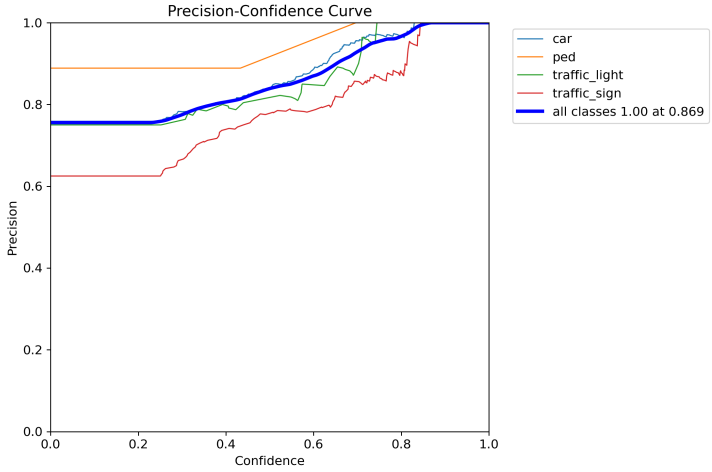

Figure 46. Precision rate in the fourth approach in sunny and good image dataset (150 EPC, BS64, No AUG and YOLOv8S in test dataset).

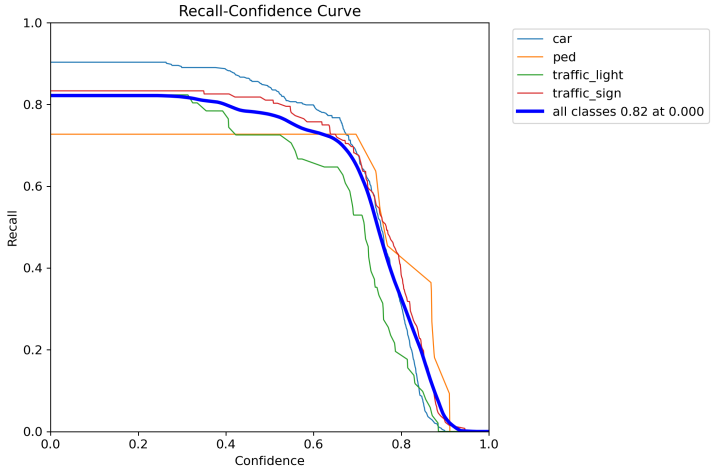

Figure 47. Recall rate in the fourth approach in sunny and good image dataset (150 EPC, BS64, No AUG and YOLOv8S in test dataset).

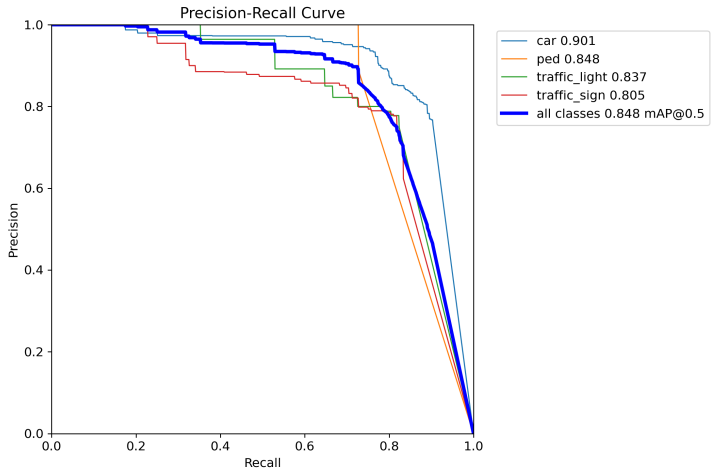

Figure 48. Precision–recall rate in the fourth approach in sunny and good image dataset (150 EPC, BS64, No AUG and YOLOv8S in test dataset).

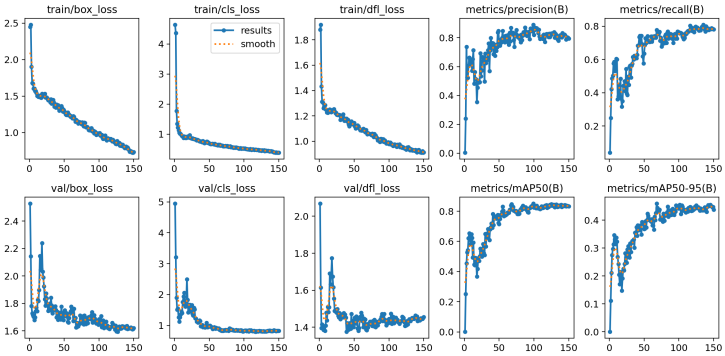

Figure 49. The final report in the fourth approach in sunny and good image dataset (150 EPC, BS64, No AUG and YOLOv8S in test dataset).

In Table 8, the precision, recall and mAP50 metrics are presented for the good or sunny test image dataset, which contains approximately 400 images. These metrics are shown across all four strategies used on the foggy dataset, which comprises about 4000 images. Unfortunately, in the final method, overfitting occurred when we used 150 epochs, a batch size of 64, no augmentation and the YOLOv8S model. This overfitting is likely due to the insufficient number of images, a problem that becomes more pronounced as the number of epochs increases.

Table 8. Precision, recall and mAP50 metrics for four proposed methods over good or sunny test dataset.

Methods	Precision %	Recall %	mAP50 % (Average for All Desired Objects)
100 EP, BS 16, NO AUG, YOLOv8S	85.71	78.08	84.25
100 EP, BS 16, AUG, YOLOv8L	82.26	74.18	81.14
100 EP, BS 64, No AUG, YOLOv8S	82.91	73.27	82.85
150 EP, BS 64, No AUG, YOLOv8S	79.48	78.08	83.17

In Table 9, there is a comparison between the object detection ratio in foggy and good weather images based on the proposed model. We benefited from averaging all object detection precision–recall metric ratios in the four proposed methods which are placed in the Table 9 first row. In the second row, we utilize data in the overall section in Table 9. It is

apparent that the object detection ratio in the test image sample dataset (sunny and good weather images dataset) is lower than it is in the foggy image dataset.

Table 9. The predicted object detection accuracy rate in foggy and good weather conditions in the proposed model.

Dataset Name	Car %	Pedestrian %	Street Sign %	Traffic Sign %
Foggy image dataset (has about 4000 images)	87.47	88.8	79.12	81.87
Sunny and good images test sample dataset (has about 400 images)	77.23	67.50	71.35	67.98

We have a comparison between YOLOv8 and YOLOv5 based on the study. We leveraged small and large versions of the YOLO model for the comparison. We tried to use the same configuration for the four methods used in the study. The comparison can be seen in Table 10. Table 10 shows that outcomes in YOLOv8 outperform YOLOv5 in the precision, recall and mAP50 metrics in the foggy dataset.

Table 10. Precision, recall and mAP50 metrics in YOLOv8 compared with the metrics in YOLOv5 for the foggy image dataset.

Models	Methods	Precision (%)	Recall (%)	mAP50 (%)
YOLOv8	100 EP, BS 16, NO AUG, YOLOv8S	82.0	73.7	83.7
YOLOv5	100 EP, BS 16, NO AUG, YOLOv5S	70.92	60.83	62.16
YOLOv8	100 EP, BS 16, AUG, YOLOv8L	84.3	78.5	83.9
YOLOv5	100 EP, BS 16, AUG, YOLOv5L	82.86	73.50	78.45
YOLOv8	100 EP, BS 64, NO AUG, YOLOv8S	81.2	72.95	83.70
YOLOv5	100 EP, BS 64, NO AUG, YOLOv5S	72.51	58.40	61.55
YOLOv8	150 EP, BS 64, NO AUG, YOLOv8S	82.3	74.3	84.0
YOLOv5	150 EP, BS 64, NO AUG, YOLOv5S	74.10	60.15	63.93

EP: Epochs, BS: Batch Size, AUG: Augmentation, S: Small, L: Large.

In the good and sunny image dataset, a comparison between the two models is performed. We also utilize small and large versions of the YOLO model for YOLOv8 and YOLOv5. The same configuration is leveraged for the four methods used in the study. The outcomes are shown in Table 11. Some results in YOLOv5 are lower as the number of images in the good or sunny dataset is about 400. In the case in which augmentation is not used, the objects are not detected correctly and the outcomes are lower.

Table 11. Precision, recall and mAP50 metrics in YOLOv8 compared with those in YOLOv5 over the good image dataset.

Models	Methods	Precision (%)	Recall (%)	mAP50 (%)
YOLOv8	100 EP, BS 16, NO AUG, YOLOv8S	85.71	78.08	84.25
YOLOv5	100 EP, BS 16, NO AUG, YOLOv5S	59.50	40.76	41.26
YOLOv8	100 EP, BS 16, AUG, YOLOv8L	82.26	84.18	81.14
YOLOv5	100 EP, BS 16, AUG, YOLOv5L	62.01	64.65	64.33
YOLOv8	100 EP, BS 64, NO AUG, YOLOv8S	82.91	73.27	82.85
YOLOv5	100 EP, BS 64, NO AUG, YOLOv5S	47.27	39.93	38.40
YOLOv8	150 EP, BS 64, NO AUG, YOLOv8S	79.48	78.08	83.17
YOLOv5	150 EP, BS 64, NO AUG, YOLOv5S	67.73	39.17	40.84

EP: Epochs, BS: Batch Size, AUG: Augmentation, S: Small, L: Large.

The required object rates are reported in Table 12 over the foggy and good or sunny image datasets. As can be seen, a comparison exists between YOLOv8 and YOLOv5, showing that YOLOv8 is better at object detection than YOLOv5.

Table 12. The average predicted object detection accuracy rate in foggy and good weather conditions in YOLOv8 compared with YOLOv5 based on the four methods used.

Models	Dataset Name	Car %	Pedestrian %	Street Sign %	Traffic Light%
YOLOv8	Foggy	87.47	88.8	79.12	81.87
YOLOv5	Foggy	69.75	73.70	56.37	67.08
YOLOv8	Sunny or Good	77.23	67.50	71.35	67.98
YOLOv5	Sunny or Good	68.42	26.4	52.32	45.77

6. Conclusions

This paper presents a methodology to recognize some objects such as cars, pedestrians, traffic lights and street signs in images that have worse weather conditions like foggy weather conditions. We propose an approach based on the YOLOv8 method for detecting objects enhancing the object detection accuracy rate. Detecting objects in worse weather conditions is challenging for self-driving vehicles.

We propose and test the model on a dataset of 4000 foggy road images changing some hyperparameters such as epochs, batch size and augmentation methods. We present four methods and we compare them for object detection in different weather conditions. The results show that the proposed techniques work very well in particular in bad weather conditions. There are some limitations in the study that are related to object detection. In some cases, the augmentation method leads to overfitting and the outcomes will have some errors. Also, increasing epochs results in overfitting as well. Some small objects in the image dataset might not be detected by some model because small objects do not have details. In the study, in the first image dataset, we do not have good and sunny images. As a result, we were forced to utilize three good and sunny image datasets to apply and analyze them on the proposed model comprehensively.

Some autonomous vehicle companies are able to benefit from other technologies such as Lidar, GPS, ultrasonics and other mobility sensors in their products at the same time; this results in achieving better outcomes to detect objects on the street. Also the sensors can be integrated with smart traffic lights or traffic signs in order to build a development in a smart city as well. An autonomous car can use more than one sensor to detect objects on roads. This leads to having fewer errors and an accurate model in identifying objects on the streets in worse situations like low visibility or high-speed objects. In addition, there is a terrific potential to benefit from more sensors in autonomous vehicles in order to recognize objects in real-time or complex environments. Future works will test the proposed model based on YOLOv8 on other datasets and images captured by autonomous vehicles.

Author Contributions: Conceptualization, A.M.M.; methodology, A.E.A.; software, A.E.A.; validation, A.M.M.; formal analysis, A.E.A.; investigation, A.E.A.; resources, A.E.A.; data curation, A.E.A.; writing—original draft preparation, A.E.A.; writing—review and editing, A.M.M.; visualization, A.E.A.; supervision, M.P.F.; project administration, A.M.M.; funding acquisition, M.P.F. All authors have read and agreed to the published version of the manuscript.

Funding: Research supported by the IN2CCAM project (EU Horizon 2020 research and innovation programme grant agreement No. 101076791). This manuscript reflects only the authors'views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

Data Availability Statement: We used four different datasets. The first, foggy road image dataset which is available on https://people.ee.ethz.ch/~csakarid/Model_adaptation_SFSU_dense/, accessed on 11 September 2024. The three good and sunny image datasets including: front and rear images of car which is available on https://www.kaggle.com/datasets/kushkunal/front-and-rear-images-of-car, accessed on 11 September 2024, and Semantic Segmentation Makassar (IDN) Road which is available on https://www.kaggle.com/datasets/nublanazqalani/semantic-segmentation-makassaridn-road-dataset, accessed on 11 September 2024, and traffic signs datasets in YOLO format whichis available on https://www.kaggle.com/datasets/valentynsichkar/traffic-signs-dataset-in-yolo-format, accessed on 11 September 2024.

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

References

1. Li, Y.; Fan, Q.; Huang, H.; Han, Z.; Gu, Q. A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition. *Drones J. Innov. Urban Mobil.* **2023**, *7*, 304. [CrossRef]
2. Islam, R.B.; Akhter, S.; Iqbal, F.; Rahman, M.S.U.; Khan, R. Deep learning based object detection and surrounding environment description for visually impaired people. *Heliyon* **2023**, *9*, e16924. [CrossRef] [PubMed]
3. Sharma, T.; Debaque, B.; Duclos, N.; Chehri, A.; Kinder, B.; Fortier, P. Deep Learning-Based Object Detection and Scene Perception under Bad Weather Conditions. *Electronics* **2022**, *11*, 563. [CrossRef]
4. Chen, X.; Chang, C.; Yu, C.; Chen, Y. A Real-Time Vehicle Detection System under Various Bad Weather Conditions Based on a Deep Learning Model without Retraining. *Sens. J. Phys. Sens.* **2020**, *20*, 5731. [CrossRef]
5. Santhanalakshmi1, S.T.; Khilar, R. A custom deep convolutional neural network cdnn—(with yolo v3 based newly constructed backbone) for multiple object detection. *J. Data Acquis. Process.* **2023**, *38*, 1511–1526.
6. Vitas, D.; Tomic, M.; Burul, M. Traffic Light Detection in Autonomous Driving Systems. *IEEE Consum. Electron. Mag.* **2020**, *9*, 90–96. [CrossRef]
7. Marode, A.; Ambadkar, A.; Kale, A.; Mangrudkar, T. Car detection using yolo algorithm. *Int. Res. J. Mod. Eng. Technol. Sci.* **2021**, *3*, 939–942.
8. Kalva, A.R.; Chelluboina, J.S.; Bharathi, D.B. Smart Traffic Monitoring System using YOLO and Deep Learning Techniques. In Proceedings of the 7th International Conference on Trends in Electronics and Informatics, Tirunelveli, India, 11–13 April 2023; pp. 831–837.
9. Lia, B.; Chen, Y.; Xu, H.; Zhong, F. Fast vehicle detection algorithm based on lightweight YOLO7-tiny. *Comput. Vis. Pattern Recognit.* **2023**, *3*.
10. Pan, W.; Chen, Y.; Liu, B. Traffic Light Detection for Self-Driving Vehicles Based on Deep Learning. In Proceedings of the 15th International Conference on Computational Intelligence and Security (CIS), Macao, China, 13–16 December 2019; pp. 63–67.
11. Iftikhar, S.; Asim, M.; Zhang, Z.; Muthanna, A.; Chen, J. Target Detection and Recognition for Traffic Congestion in Smart Cities Using Deep Learning-Enabled UAVs: A Review and Analysis. *Appl. Sci.* **2023**, *13*, 3995. [CrossRef]
12. Wang, J.; Chen, C.; Wang, C.W. Street Sign Recognition Algorithm Based on Deep Learning. In Proceedings of the 3rd International Conference on Image and Graphics Processing (ICIGP 20), Singapore, 8–10 February 2020; pp. 31–35.
13. Diaz, M.; Cerri, P.; Pirlo, G.; Ferrer, M.A.; Impedovo, D. A Survey on Traffic Light Detection. *Int. Conf. Image Anal. Process.* **2015**, *9281*, 201–208.
14. Liu, Z.; Li, D.; Ge, S.S.; Tian, F. Small traffic sign detection from large image. *Appl. Intell. J.* **2019**, *50*, 1–13. [CrossRef]
15. Khan, N.; Bhalerao, S. A Review Paper On Different Type of Object Detection Techniques for Vehicle. *Int. Res. J. Mod. Eng. Technol. Sci.* **2022**, *4*, 1068–1072.
16. Tsai, V.J.D.; Chen, J.-H.; Huangn, H.-S. Traffic sign inventory From google street view images. *Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *41*, 243–246.
17. Yorozu, Y.; Hirano, M.; Oka, K.; Tagawa, Y. A Comparative Study between State-of-the-Art Object Detectors for Traffic Light Detection. In Proceedings of the International Conference on Emerging Trends in Information Technology and Engineering, Vellore, India, 24–25 February 2020; pp. 1–6.
18. Liu, X.; Zhang, B.; Liu, N. CAST-YOLO: An Improved YOLO Based on a Cross-Attention Strategy Transformer for Foggy Weather Adaptive Detection. *Appl. Sci. J.* **2023**, *13*, 1176. [CrossRef]
19. Gochoo, M.; Otgonbold, M.; Ganbold, E.; Hsieh, J.; Chang, M.; Chen, P.; Dorj, B.; Jassmi, H.A.; Batnasan, G.; Alnajjar, F.; et al. FishEye8K: A Benchmark and Dataset for Fisheye Camera Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Vancouver, BC, Canada, 17–24 June 2023; pp. 5305–5313.
20. Han, C.; Gao, G.; Zhang, Y. Real-time small traffic sign detection with revised Faster R-CNN. *Multimed. Tools Appl.* **2019**, *78*, 13263–13278. [CrossRef]
21. Possatti, L.C.; Guidolini, R.; Cardoso, V.B.; Berriel, R.F.; Paixão, T.M.; Badue, C.; Souza, A.F.D.; Oliveira-Santos, T. Traffic Light Recognition Using Deep Learning and Prior Maps for Autonomous Cars. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
22. Ashwini, A.; Purushothaman, K.E.; Prathaban, B.P.; Jenath, M.; Prasanna, R. Automatic Traffic Sign Board Detection from Camera Images Using Deep learning and Binarization Search Algorithm. In Proceedings of the International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI), Chennai, India, 19–21 April 2023; pp. 1–5.
23. Yu, X.; Si, Y.; Li, L. Pedestrian detection based on improved Faster RCNN algorithm. In Proceedings of the IEEE/CIC International Conference on Communications in China (ICCC), Changchun, China, 11–13 August 2019; pp. 346–351.
24. Dong, D.V. Application of Advanced Deep Convolutional Neural Networks for the Recognition of Road Surface Anomalies. *Eng. Technol. Appl. Sci. Res. J.* **2023**, *13*, 10765–10768.
25. Tasyurek, M.; Gul, E. A new deep learning approach based on grayscale conversion and DWT for object detection on adversarial attacked images. *J. Supercomput.* **2023**, *79*, 20383–20416. [CrossRef]

26. Liu X.; Lin, Y. YOLO-GW: Quickly and Accurately Detecting Pedestrians in a Foggy Traffic Environment. *Sens. J. Environ. Sens.* **2023**, *23*, 5539. [CrossRef]
27. Khalid, S.; Oqaibi, H.M.; Aqib, M.; Hafeez, Y. Small Pests Detection in Field Crops Using Deep Learning Object Detection. *Sustain. J. Sustain. Technol. Improv. Soil Crop. Environ. Qual. Chang. Clim.* **2023**, *15*, 6815. [CrossRef]
28. Janahiraman, T.V.; Subuhan, M.S.M. Traffic Light Detection Using Tensorflow Object Detection Framework. In Proceedings of the IEEE 9th International Conference on System Engineering and Technology (ICSET), Shah Alam, Malaysia, 7 October 2019; pp. 108–113.
29. Islam, K.T.; Wijewickrema, S.; Raj, R.G.; O'Leary, S. Street Sign Recognition Using Histogram of Oriented Gradients and Artificial Neural Networks. *J. Imaging Trends Mach. Learn. Vis. Comput.* **2019**, *5*, 44. [CrossRef]
30. Ta¸syürek, M. A New Approach For Spatial Street Sign Detection from EXIF Using Deep Learning-based Object Detection, Distance Estimation, Rotation and Projection System. *Vis. Comput. J.* **2023**, *40*, 983–1003. [CrossRef]