

Article

Improving the Perception of Objects Under Daylight Foggy Conditions in the Surrounding Environment

Mohamad Mofeed Chaar , Jamal Raiyn  and Galia Weidl 

University of Applied Sciences Aschaffenburg, 63743 Aschaffenburg, Germany; jamal.raiy@th-ab.de (J.R.); galia.weidl@th-ab.de (G.W.)

* Correspondence: mohamadmofeed.chaar@th-ab.de

Abstract: Autonomous driving (AD) technology has seen significant advancements in recent years; however, challenges remain, particularly in achieving reliable performance under adverse weather conditions such as heavy fog. In response, we propose a multi-class fog density classification approach to enhance the AD system performance. By categorizing fog density into multiple levels (25%, 50%, 75%, and 100%) and generating separate datasets for each class using the CARLA simulator, we improve the perception accuracy for each specific fog density level and analyze the effects of varying fog intensities. This targeted approach offers benefits such as improved object detection, specialized training for each fog class, and increased generalizability. Our results demonstrate enhanced perception of various objects, including cars, buses, trucks, vans, pedestrians, and traffic lights, across all fog densities. This multi-class fog density method is a promising advancement toward achieving reliable AD performance in challenging weather, improving both the precision and recall of object detection algorithms under diverse fog conditions.

Keywords: object recognition; severe weather; CARLA simulation; bounding box; computer vision; autonomous driving



Citation: Chaar, M.M.; Raiyn, J.; Weidl, G. Improving the Perception of Objects Under Daylight Foggy Conditions in the Surrounding Environment. *Vehicles* **2024**, *6*, 2154–2169. <https://doi.org/10.3390/vehicles6040105>

Academic Editor: Richard Romano

Received: 3 October 2024

Revised: 23 November 2024

Accepted: 1 December 2024

Published: 18 December 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Autonomous driving (AD) is based on the principle of driving vehicles using artificial control and perception without human intervention [1]. Autonomous driving systems use a variety of sensors to perceive their surroundings [2], including cameras [3], radar [4], and LiDAR [5]. These sensors provide the system with information about the location of other vehicles, pedestrians, and objects in the environment. The system then uses this information to make decisions about how to control the vehicle. Autonomous driving systems are still under development, and they carry a promise with the potential to revolutionize transportation. AD could make transportation safer [6], more efficient, and accessible.

However, AD still faces challenges and open problems such as perception under severe weather conditions.

Numerous studies have investigated perception under foggy conditions. However, these studies have generally treated fog as a binary classification problem (foggy vs. non-foggy) and extrapolated conclusions to various levels of fog density. This approach overlooks the need for improved perception tailored to each specific fog density category.

Our research proposes a novel approach to object detection in foggy conditions employing a data-driven strategy and machine learning techniques. We categorize fog density into five distinct levels: 0%, 25%, 50%, 75%, and 100% (see Appendix A). Leveraging the CARLA simulator (Car Learning to Act) [7,8], we generate a comprehensive dataset encompassing a diverse range of fog densities [9]. Subsequently, we implement a bounding box-based machine learning algorithm to effectively detect objects under varying fog conditions. and we obtained highly accurate results for all fog levels, from clear weather to the highest fog density (100%). The purpose of this work is to enhance object recall (alongside

precision) in multiple categories of fog conditions. We achieved high recall and precision across all fog density levels.

Given the close relationship between this research and safety-critical applications such as autonomous driving, examining the potential impact on navigation and vehicle safety is essential. Emphasizing how this model could be integrated into existing vehicle systems or improve object recognition accuracy in foggy conditions could significantly enhance the research's relevance and practical application in real-world autonomous systems.

This paper is organized as follows. Section 2 gives an overview of the influence of the wearer on the performance of autonomous vehicles. The next sections describe the methodology (Section 3), discuss the results (Section 4), and conclude the discussion as well as point out directions for future research (Section 5).

2. Related Work

Weather phenomena can have various negative influences on the performance of autonomous vehicles (AVs), especially in their perception and sensing systems. Adverse conditions like heavy rain, snow, fog, and low lighting can significantly impair the sensors that AVs rely on, such as cameras, radar, LiDAR, and ultrasonic sensors. These systems are crucial for detecting obstacles, lane markings, pedestrians, and other vehicles. The diminished performance in such conditions poses a serious challenge to AV safety and reliability [10].

The authors (Diaz-Ruiz et al., 2022) [11] have developed datasets specifically tailored for severe weather conditions, including cloudy, rainy, snowy, night, and sunny scenarios. These datasets were generated using multiple sensors, and the data for each weather condition were trained separately. This approach significantly enhanced perception and increased accuracy. The authors demonstrated that models trained for specific weather conditions yield more accurate object detection when applied in those same conditions. For example, the model trained on data from sunny conditions achieved a mean average precision (mAP_{.5: 0.95}) of 54.3 when tested under sunny conditions but only 38.9 when tested in rainy weather. Conversely, the model trained on rainy weather data yielded an (mAP_{.5: 0.95}) of 46.3 in rainy conditions, showing an improvement in accuracy from 38.9 to 46.3. However, this approach did not include the foggy conditions. In our work, we focused specifically on foggy conditions, dividing them into four distinct classes in addition to sunny conditions. We utilized the CARLA simulation environment to generate the datasets and employed our filtering techniques within the CARLA simulator to accurately label the data [12] and we achieved 0.739 in heavy fog (mAP_{.5: 0.95}).

Furthermore, Valanarasu et al. (2022) [13] proposed a transformer-based model to restore images degraded by adverse weather conditions. The authors argue that transformers can be adapted to image restoration by treating images as sequences of pixels. The proposed model, called Transweather, consists of an encoder and a decoder [14,15]. The encoder takes an image degraded by adverse weather conditions as input and produces a latent representation of the image. The decoder then takes the latent representation as input and produces a restored image. The encoder is a multilayer convolutional transformer (MCT) model. The MCT model consists of a stack of convolutional layers and encoder-decoder attention layers. The convolutional layers extract features from the image, while the attention layers allow the model to learn long-range dependencies between pixels. The decoder is a convolutional transformer decoder (CTD) model. The CTD model consists of a stack of decoder-encoder attention layers and upsampling layers. The attention layers allow the model to attend to the latent representation of the image, while the upsampling layers reconstruct the restored image. The authors evaluated Transweather on a dataset of images degraded by rain, snow, haze, and fog. The results showed that Transweather outperforms several state-of-the-art image restoration methods. In previous work, fog was categorized as a single class (fog or no fog), which posed challenges, particularly when dealing with light fog. In contrast, our approach did not utilize TransWeather to transform foggy images into non-foggy ones. Instead, we focused on enhancing perception

directly within foggy conditions. We developed separate models tailored to different fog densities, ranging from light to heavy fog, to improve accuracy and robustness across varying fog intensities.

The authors (Bijelic et al., 2020) [16] introduced an innovative approach by integrating four sensors—an RGB camera, LiDAR, a gated camera, and radar—into a unified perception system. The outputs of these sensors were projected into the camera’s coordinate space and then processed through a convolutional neural network with four input channels to enhance perception accuracy. The authors evaluated their method using a benchmark dataset focused on object detection in adverse weather conditions. Their approach was compared against several state-of-the-art single-sensor and fusion methods. The results demonstrated that their method outperformed existing approaches, achieving an average precision of 76.69 in heavy fog. In comparison, our approach further improved performance, achieving an average precision of 89.00. We cannot compare the two results directly due to the differing data types (simulation vs. real data), although the weather conditions are the same.

The authors (Li et al., 2023) [17] propose a domain adaptation framework that leverages both labeled data from the source domain (clear weather) and unlabeled data from the target domain (foggy weather). The key components of their approach include feature alignment, which involves mechanisms to align the feature distributions between the clear and foggy weather domains, helping the model to learn domain-invariant features that are robust to weather changes. They also employ domain adversarial training, using a domain discriminator to distinguish between the source and target domains, where the object detector is trained adversarially to perform well in both domains by confusing the discriminator, leading to features that generalize across different weather conditions. Additionally, the paper proposes multi-level adaptation, where adaptation occurs at multiple levels of the detection pipeline, including both the image and feature levels, to enhance the model’s robustness to foggy conditions. They also incorporate a self-training mechanism where the model iteratively generates pseudo-labels for the foggy images and refines its predictions, allowing the model to learn from the target domain data without requiring explicit labels. The mean average precision mAP for this work is 42.3 for heavy fog, 36.5 for walkers, and 50 for detecting walkers under heavy fog up to 200 m in distance.

The paper “A Review of the Impacts of Defogging on Deep Learning-Based Object Detectors in Self-Driving Cars” (Ogunrinde & Bernadin, 2021) [18] explores the effects of image defogging techniques on the performance of deep learning-based object detection systems used in autonomous vehicles. The authors analyze the effectiveness of these techniques in improving detection accuracy, highlighting that while defogging generally enhances image quality, its impact on detection performance varies depending on the method used. Some defogging approaches may introduce artifacts or alter important features in the images, potentially leading to reduced detection accuracy or false positives. The paper emphasizes the need for the careful selection and tuning of defogging methods to balance the trade-off between improved visibility and accurate object detection. Additionally, the authors discuss the potential of integrating defogging directly into the object detection pipeline, allowing models to learn defogging and detection tasks simultaneously. Using their methodology, they improved recall under heavy fog conditions from 59.61 to 62.02 and precision from 60.98 to 62.74. In comparison, our approach resulted in a more significant increase with recall improving from 43.4 to 63.6 and precision from 86.8 to 93.1. The differences in recall between our results and theirs can be attributed to the variations in the datasets used and the algorithms implemented; we employed YOLOv8 [19], while they used YOLOv3 [20].

The overviewed papers have generally treated fog as a binary class (fog or no fog). In contrast, our research introduces a more nuanced approach by developing four distinct categories for fog density (besides clear weather) with a separate model implemented for each category. Our findings demonstrate that by categorizing fog into multiple levels, we can significantly enhance perception accuracy compared to the binary classification

approach. This methodology can be adopted by the studies mentioned above to improve their perception accuracy and achieve more precise results.

In this work, our objective is to improve perception under heavy fog conditions. Our novelty involved classifying fog levels into four distinct categories based on fog density, besides the clear weather (0% or clear weather, 25%, 50%, 75%, and 100%). We then train the model by data, using deep learning techniques tailored for object detection. The method's foundation lies in categorizing first the input based on the fog density and then operating a model specifically trained for that particular fog density range (refer to Figure 1). For the dataset, we employed the CARLA simulator, which allows us to precisely control fog density and gather data with automated labeling for object detection in foggy conditions. We have made the data collection project available on our GitHub (<https://github.com/Mofeed-Chaar/Improving-bounding-box-in-Carla-simulator>, accessed on 2 October 2024) [18]. Additionally, we implemented flexible weather control by modifying parameters within the YAML file [21] named *weather.yaml* within our GitHub project. The objects we focused on in our work comprise six distinct categories: cars, buses, trucks, vans, pedestrians, and traffic lights. Furthermore, we meticulously generated distinct datasets for each fog density level and trained individual object detection models for each class of fog density. This approach yielded consistently high results across various metrics, including precision, recall, and MAP@50. In particular, we achieved an accuracy of more than 90% under a heavy fog condition (100% fog density), as we will see later in this paper.

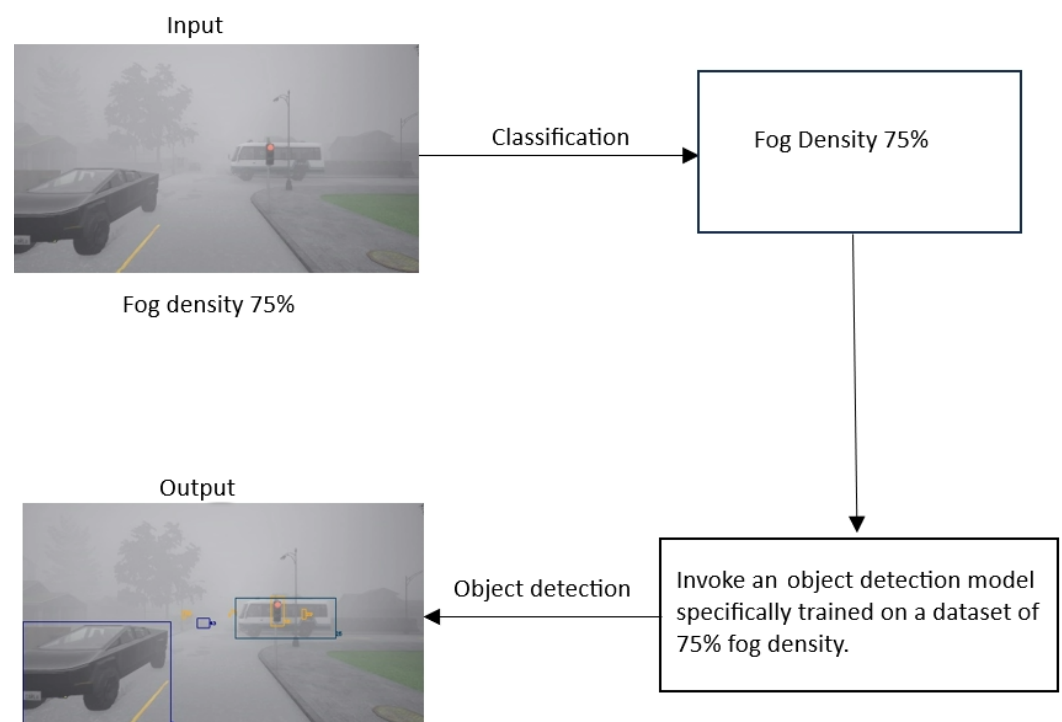


Figure 1. For a start, determine the fog density of the input image, then use a model specifically trained for that fog density level. In this example, the fog density is 75%.

3. Methodologies

3.1. Generate Data

The dataset plays a critical role in the development and enhancement of algorithms, as the quality and appropriateness of the data are foundational to the success of machine learning models. To ensure diverse and comprehensive data, collection can be conducted through various methods.

One approach involves capturing real-world data by driving on public roads, allowing for the recording of natural driving conditions and variability in weather, lighting, and traffic patterns [22–25]. This method is beneficial for gathering authentic data that reflect

true environmental and road conditions, which is essential for training robust machine learning models.

Another approach includes data collection in controlled laboratory settings [26]. Laboratory environments allow for the precise manipulation of variables such as lighting, object placement, and sensor calibration, which helps isolate specific factors influencing model performance. Controlled settings can be particularly valuable for fine-tuning algorithms under known conditions or testing edge cases that may not frequently occur in real-world driving.

In addition to these traditional data collection methods, synthetic datasets can be generated, providing a flexible and scalable alternative for training machine learning models. Synthetic data can be created by adding fog or other environmental factors to clear-weather images [27], enabling the simulation of various weather conditions without physically capturing them. Furthermore, simulation environments, such as the CARLA simulator, offer an advanced platform for generating synthetic data. CARLA allows researchers to not only simulate fog but also precisely control its density, which aids in creating datasets that reflect a range of visibility conditions. This capability to manipulate environmental factors provides researchers with flexible, high-quality data tailored to specific needs and scenarios, supporting the development of models that perform well in diverse and challenging conditions.

Additionally, simulations can be used to generate data, with tools like the CARLA simulator offering a cost-effective and flexible way to control environmental conditions. Simulated data are generally less expensive and allow for precise adjustments to variables like weather and visibility. However, the quality of simulated data typically does not match that of real-world data, as it may lack certain complexities and nuances present in actual driving environments.

3.2. CARLA Simulator

CARLA (Car Learning to Act) [8] is a high-fidelity, open-source simulator widely used in autonomous driving research [7]. It provides a detailed, realistic urban environment complete with various road types, buildings, vehicles, and pedestrian models, making it an ideal platform for developing and testing algorithms for self-driving vehicles. CARLA's environment includes intersections, traffic lights, roundabouts, and a range of obstacles that mirror real-world conditions, thereby enabling researchers to simulate complex driving scenarios.

One of CARLA's key advantages is its ability to simulate and control environmental variables, including weather and lighting. This flexibility is particularly useful for generating datasets under specific weather conditions, such as fog, rain, or varying times of day. Using CARLA, researchers can create datasets with automatically labeled 3D or 2D bounding boxes within these controlled conditions [28]. This automated labeling is efficient and time saving, as it circumvents the manual annotation process typically required for training data. For our study, CARLA served as a critical tool in generating a dataset with multiple fog density classes (clear, 25%, 50%, 75%, 100%). Obtaining real-world data with these precise fog levels would be challenging, as capturing consistent fog densities on public roads is impractical, and obtaining representative images from existing sources is limited. Moreover, the task of labeling bounding boxes in dense fog (100% fog density) presents an additional difficulty, as thick fog often obscures or partially hides objects, making manual labeling highly challenging. CARLA's controlled environment overcomes these limitations, allowing us to produce a comprehensive dataset tailored to our needs while providing accurately labeled bounding boxes across all fog density levels. This dataset forms the foundation of our work, enabling us to develop and test object detection models under varying fog conditions that simulate real-world challenges for autonomous driving systems.

3.3. Yolo (You Only Look Once)

A Convolutional Neural Network (CNN) represents a specialized category within artificial intelligence that focuses on analyzing input data with inherent spatial structures.

Regarded as a pivotal component of AI, CNNs employ interconnected computational elements (neurons) to process perceptual data derived from the surrounding environment. CNNs serve as a subset of deep learning models capable of handling one-dimensional, two-dimensional, and three-dimensional data. Their primary purpose is to discern spatial hierarchies of features autonomously and adaptively, progressing from low- to high-level patterns [29]. They are typically comprising three convolutional layers, pooling, and a fully connected layer—CNNs utilize convolutional and pooling layers for feature extraction, while the fully connected layer maps the extracted features to produce the final output, such as classification. The CNN architecture encompasses three distinctive layers: convolutional, pooling, and classification. The convolutional layers serve as the heart of the CNN, where weights define a convolutional kernel applied to the original input in small, incremental receptive fields [30]. YOLO [31] is a real-time object detection algorithm that divides an image into a grid and predicts the bounding boxes and class probabilities for each object in the grid using CNN. It is a popular algorithm for object detection because it is fast, accurate, and easy to use. Due to its remarkable capabilities, YOLO has found widespread application in autonomous driving systems [32]. Moreover, there have been numerous versions of YOLO, each striving to enhance its accuracy and reduce latency, such as YOLOv5 [33] and YOLOv8 [19]. The loss function in YOLO is determined as the following equation [31]:

$$\begin{aligned} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{K}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \\ & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{K}_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] + \\ & \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{K}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{K}_{ij}^{noobj} (C_i - \hat{C}_i)^2 + \\ & \sum_{i=0}^{S^2} \mathbb{K}_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \end{aligned} \quad (1)$$

where $\mathbb{K}_{ij}^{obj}$ equals 1 if the object appears in cell i with box number j ; otherwise, it will be zero, S is the cell, B is the anchor box, (x_i, y_i, w_i, h_i) is $(x_{center}, y_{center}, width, height)$, respectively, in the base of the box.

Metrics

In machine learning, precision and recall are two important metrics used to evaluate the performance of a classifier. They are commonly used in tasks such as spam filtering, fraud detection, and medical diagnosis. Precision [2] measures the accuracy of positive predictions. It represents the proportion of positive predictions that are actually correct. High precision indicates that the classifier does not make many false positives. Formally, precision is defined as the following:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

where TP, FP are true positives and false positives, respectively.

Recall [34], also known as sensitivity, measures the completeness of positive predictions. It represents the proportion of actual positive instances that were correctly identified by the classifier. A high recall indicates that the classifier does not miss many true positives. Formally, recall is defined as

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

where FN is a false negative.

Precision and recall often have an inverse relationship. In other words, increasing one metric often comes at the expense of the other. This is because a classifier that is very strict with respect to its positive predictions may miss some true positives, resulting in a lower recall. Conversely, a classifier that is more lenient may identify more true positives, but it may also increase the number of false positives, leading to a lower precision. To address the trade-off between precision and recall [35], the F1 score [36] is often used. It is a harmonic mean of precision and recall, which gives equal weight to both metrics. A high F1 score indicates that the classifier performs well in both aspects. The equation of F1 is described as follows:

$$F1_Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

The F1 score is a useful metric for evaluating the overall performance of a classifier. It provides a single measure that captures both precision and recall, allowing for a more balanced assessment of the classifier's performance. The other metric is the mean average precision mAP50 [37], where 50 describes the intersection over union IOU. The mAP50 value is calculated by averaging the precision–recall curves (PRCs) for each object class in the dataset. A PRC is a plot of precision against recall, where each point on the curve corresponds to a given IoU threshold. The area under the curve (AUC) is used to measure the overall performance of the model. In general, a higher mAP50 indicates that the object detection model has better performance in detecting objects with a high degree of confidence. This is important for tasks such as object tracking, image segmentation, and autonomous vehicles, where accurate object detection is crucial for reliable and safe operation.

3.4. Collect the Data

Acquiring datasets that meet our specific requirements proved challenging due to the need for a diverse range of fog density levels. To address this obstacle, we opted to employ simulation to generate datasets encompassing five distinct fog conditions, which are categorized by fog density (clear, 25%, 50%, 75%, and 100%). We integrated the simulation to automatically label the bounding boxes (refer Figure 2) for six objects (car, bus, truck, van, walker, and traffic light) using eight maps within the CARLA simulator. Our data collection process entailed the following steps:

- Establish an environment with fog conditions adhering to our specifications and designate the map name.
- Remove all vehicles from the parking lot as they cannot be effectively labeled as bounding boxes due to the simulator's inability to automate the labeling process for parked vehicles.
- Mount the sensor on a random vehicle, designate it as the ego vehicle and enable autonomous driving.
- Gather data from the sensors by capturing several RGB sensor images and preserving them accompanied by bounding box annotations.

Through our data collection process, we have amassed a comprehensive dataset of 40,000 images for each of five distinct fog density categories with each map contributing 5000 images. This substantial dataset provides a rich resource for training and evaluating fog-aware autonomous driving algorithms. We meticulously categorized our dataset into four distinct distance ranges labeling all objects up to 50 m, up to 100 m, up to 150 m, and up to 200 m, and saved the labels as a text file. Each image within these ranges was thoroughly labeled with the corresponding objects present within the scene. To ensure consistency and clarity, we have saved a consistent image resolution (1280 × 720 pixels) for all images. Additionally, we extended our datasets by incorporating data from other sensors, such as LiDAR, radar, semantic segmentation images, and depth camera images. Furthermore, the weather parameters for all data that we have generated for our study were detected and presented in Table 1 [38], the difference is only in fog density. This comprehensive dataset provides a valuable resource for researchers and developers working on autonomous

driving algorithms. since it allows the handling of various fog conditions and various sensor modalities, and it is available in our GitHub project.

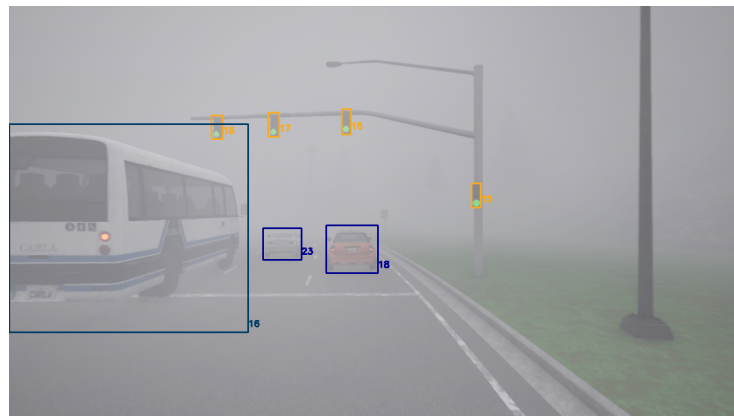


Figure 2. An RGB camera sensor in the CARLA simulator captures an image, the horizontal field of view in degrees (fov) is 90.0 degrees, and the dimensions are 1280×720 (see Appendix B) with bounding boxes overlaid on the image to identify objects within the scene. The bounding boxes are generated automatically by the simulator, and they provide a visual representation of the objects' positions and dimensions. The fog density on this image is 100%.

Table 1. The weather parameters that we generated in our data.

Weather Parameters	Value	Range	Note
Sun Azimuth	300.00	1168	
Sun Altitude	45.00	961	
Cloudiness	[20–40] ¹	[0, 100]	
Precipitation	0.00	[0, 100]	
Precipitation Deposits	0.00	[0, 100]	
Wind Intensity	10.00	[0, 100]	
Fog Density	[0, 25, 50, 75, 100] ²	[0, 100]	
Fog Distance	0.75	[0, ∞]	Visibility
Fog Falloff	0.10	[0, ∞]	Describes the dense and heavy
Wetness	0.00	[0, 100]	
Scattering Intensity [39]	1.00	[0, ∞]	Light contribution to volumetric fog
Mie Scattering [40]	0.03	[0, ∞]	Interaction of light with large particles
Rayleigh Scattering [41]	0.0331	[0, ∞]	Interaction of light with small particles
Dust Storm	0.00	[0, 100]	

¹ To increase the diversity of our dataset, we randomly generated some samples with a cloudiness of 20 and others with a cloudiness of 40. ² The fog density was a primary focus of our study, and we generated data with varying levels of fog density: 0%, 25%, 50%, 75%, and 100%. Note: All parameters are editable in our GitHub project in file weather.yaml.

4. Results and Discussion

The datasets we generated were divided into five categories, each corresponding to a specific weather condition with varying fog density. For each category of fog density, we labeled objects into four ranges: objects within 50 m, objects within 100 m, objects within 150 m, and objects within 200 m. We then trained different models using YOLOv5s and YOLOv8m for a variety of distance ranges using specific hyperparameters (refer to Table 2). For the latency of the YoloV8 model, see Table 3 [42].

Table 2. The hyperparameters that we used to optimize our model's.

Hyperparameter	Epochs	Batch ¹	IOU	Learning Rate ^{1 2}	Momentum ³	Weight Decay ⁴
Value	50	[8, 16]	0.7	0.01	0.937	0.0005

¹ 8 for training of images (1280×1280) pixels, and 16 for (640×640). ² This is the step size that the model takes toward the negative gradient of the loss function. ³ This is a parameter that helps to stabilize the training process by smoothing out the updates to the model's weights. ⁴ This is a parameter that helps to prevent the model from overfitting by penalizing large weight updates. Note: For more hyperparameters information, see Appendix C.

Table 3. The latency of our YOLOv8m model when applied to our dataset.

Image Size	Preprocess	Inference	Postprocess	Total
1280	0.3 ms	6.6 ms	0.3 ms	7.2 ms
640	0.1 ms	2.9 ms	0.3 ms	3.3 ms

In our validation process, we utilized an NVIDIA RTX 4090 GPU, an Intel Core i9-14900K processor, and 64 GB of RAM operating at a frequency of 4000 MHz.

Our training results suggest that training our models on datasets with varying fog densities can preserve their performance and even enhance their accuracy in heavy fog conditions. Our corresponding results of the training, on the base of YOLOv5s, are shown in Table 4. We are utilizing the YOLO loss function (refer to Equation (1)).

We have separated this dataset of objects, labeled within 50 m, into 80% for training and 20% for validation with an image size of 640×640 pixels.

Table 4. Accuracy of object detection for six classes on each of fog density using YOLOv5s.

Fog Density	Precision	Recall	mAP50	mAP50-95
0%	0.968	0.926	0.965	0.841
25%	0.976	0.925	0.965	0.841
50%	0.969	0.925	0.968	0.858
75%	0.958	0.861	0.924	0.777
100%	0.952	0.834	0.89	0.739

Note: 1-The optimization algorithm employed in this study is stochastic gradient descent (SGD) [43].

2-The object distances for this training were restricted to objects within 50 m.

These results represent the performance of our models across six object classes. It is important to note that the accuracy is not uniform across all classes with some classes performing better than others. This is due to a number of factors, including the shape, size, and texture of the objects, as well as the presence of other objects in the scene (refer to Table 5).

This procedure effectively preserved the precision (refer to Equation (2)) of object detection in heavy fog conditions, while the recall (refer to Equation (3)) was inversely proportional to the fog density. This trend was consistent even when the training data were expanded to include objects within longer distances, such as 100 m or more. We trained the YOLOv8m model using the same hyperparameters as the YOLOv5s model for all distances of object detection (50 m, 100 m, 150 m, 200 m) (see Table 6). This allowed us to directly compare the performance of the two models under the same conditions. We can conclude that precision remains largely unaffected when data are used beyond 50 m, but recall exhibits a decreasing trend. This can be attributed to the consistent detection of close objects, but the model's ability to identify objects at greater distances was diminished, impacting recall. We can deduce that object detection is highly accurate for close objects, but it becomes less accurate for objects with increasing distances. This is due to the fact that the fog obscures the objects, making it harder for the model to distinguish between the objects and the background.

Table 5. The accuracy of object detection for each class on each fog density is shown in the following table. YOLOv5m was used as the object detection model, and the image size was 640×640 pixels. The object detection model was trained on a dataset of images that included labels for all objects up to 50 m in distance. As demonstrated, we achieved high precision and recall for object detection in dense fog conditions by using a model specifically trained for high fog density.

Fog Density	Class	Precision	Recall	mAP50	mAP50-95
Clear weather	car	0.898	0.86	0.929	0.756
	bus	0.909	0.978	0.973	0.863
	truck	0.935	0.534	0.712	0.531
	van	0.892	0.864	0.905	0.79
	walker	0.879	0.624	0.718	0.459
	traffic light	0.958	0.92	0.951	0.809
Fog density 25%	car	0.935	0.848	0.929	0.75
	bus	0.981	0.891	0.903	0.799
	truck	0.921	0.821	0.909	0.785
	van	0.932	0.878	0.949	0.831
	walker	0.941	0.639	0.778	0.485
	traffic light	0.976	0.92	0.965	0.818
Fog density 50%	car	0.927	0.913	0.96	0.811
	bus	0.969	0.951	0.975	0.843
	truck	0.934	0.763	0.809	0.722
	van	0.934	0.893	0.945	0.807
	walker	0.933	0.791	0.867	0.591
	traffic light	0.968	0.922	0.957	0.827
Fog density 75%	car	0.94	0.865	0.936	0.776
	bus	0.968	0.867	0.94	0.831
	truck	0.96	0.845	0.899	0.813
	van	0.96	0.869	0.937	0.817
	walker	0.945	0.791	0.873	0.597
	traffic light	0.983	0.931	0.966	0.852
Fog density 100%	car	0.925	0.86	0.919	0.759
	bus	0.933	0.934	0.976	0.859
	truck	0.964	0.676	0.727	0.628
	van	0.968	0.876	0.94	0.835
	walker	0.942	0.73	0.82	0.558
	traffic light	0.986	0.923	0.96	0.81

Table 6. The accuracy of object detection range of labels distance at each fog density is shown in the following table. The YOLOv8m [19] model was used for the object detection model, and the image size was 640×640 pixels.

Distance	Fog Density	Precision	Recall	F ₁ -Score	mAP50	mAP50-95
All objects within 50 m	0%	0.947	0.805	0.87	0.882	0.763
	25%	0.927	0.858	0.891	0.922	0.822
	50%	0.953	0.883	0.916	0.938	0.839
	75%	0.959	0.868	0.911	0.93	0.837
	100%	0.941	0.719	0.815	0.824	0.723
All objects within 100 m	0%	0.915	0.664	0.769	0.75	0.614
	25%	0.924	0.692	0.791	0.787	0.645
	50%	0.914	0.716	0.803	0.803	0.66
	75%	0.945	0.668	0.782	0.758	0.633
	100%	0.928	0.637	0.755	0.735	0.59
All objects within 150 m	0%	0.888	0.581	0.702	0.664	0.526
	25%	0.921	0.606	0.731	0.699	0.563
	50%	0.887	0.579	0.7	0.668	0.538
	75%	0.903	0.549	0.682	0.631	0.519
	100%	0.909	0.453	0.604	0.525	0.425
All objects within 200 m	0%	0.89	0.525	0.66	0.611	0.476
	25%	0.893	0.552	0.682	0.461	0.5
	50%	0.901	0.531	0.668	0.618	0.486
	75%	0.917	0.483	0.632	0.56	0.452
	100%	0.912	0.44	0.593	0.516	0.413

Table 6 shows that the model can detect objects with high precision (see Figure 3).



Figure 3. We tested our object detection model in heavy fog conditions (fog density 100%) using a model that was trained with labels for all objects up to 200 m. The model trained with data under fog density 100% outperformed the model trained on clear data in detecting objects at long distances. As shown in (a), on the left, the fog-trained model successfully detected distant objects, while the clear-data model struggled to do so, as evident in (b) on the right. This difference in performance primarily increases the recall of the model. The red box describes the cars, the orange box describes the vans, and the green box describes the traffic lights.

At greater distances, the model may miss some objects, but this is acceptable given the increased difficulty of detecting objects in fog. In case of heavy fog, driving behavior and speed are significantly affected. Aside from the speed limit imposed in heavy fog conditions, drivers adapt their driving style accordingly. The priority in heavy fog is to prioritize close objects and gradually increase the perception with distance. This is because the visibility is considerably reduced in heavy fog, making it challenging to identify objects at farther distances. Our object detection model can accurately identify objects in foggy conditions even when visibility is reduced. We achieved this by training the model on a large dataset of images taken in various fog densities. Our model can detect objects with high precision under heavy fog conditions.

In our previous work, we trained our object detection model using images with a resolution of 640×640 pixels. However, we noticed that using higher resolution (1280×1280 pixels) resulted in improved recall. The results of this experiment are summarized in Table 7. These results are essential for our work, where we implemented a special model for each fog category.

Moreover, as seen in Table 7, the larger objects (e.g., buses) exhibit higher accuracy than smaller objects (e.g., walkers), particularly in terms of recall. Note that large objects face less accuracy degradation with increasing distances compared to smaller objects, and the impact of recall degradation on small objects at high distances is more pronounced than on larger objects. However, using higher resolutions, such as 1280×1280 pixels, can resolve this issue. Note that there is a trade-off between resolution and latency. To address this, we can employ an appropriate model for each fog condition. Additionally, accuracy is more crucial than latency in heavy fog conditions because vehicle speeds are slower than in clear weather. On the other hand, we found that traffic lights (as objects) are detected with high accuracy despite being small objects (see Tables 5 and 7 and Figure 3). This is likely due to the distinct features surrounding traffic lights, such as the traffic light poles, their positioning on the roadside, and the colored states of the traffic signals. Generally, the performance of our object detection model is highly accurate for fog density levels that match the fog density levels used to train the model. However, when the model is validated at fog density levels that differ from the levels used for training, the accuracy decreases (refer Table 8). As evident from Table 8, we can conclude that using a model trained for the same fog density significantly enhances the precision and recall. Notably, the highest accuracy values appear on the diagonal of the table, corresponding to the validation of models trained on the corresponding fog density categories.

Table 7. The accuracy of object detection for each class on each fog density is shown in the following table. YOLOv8m was used as the object detection model, and the image size was 1280×1280 pixels. The object detection model was trained on a dataset of images that included labels for all objects up to 200 m in distance.

Fog Density	Class	Precision	Recall	mAP50	mAP50-95
Clear weather	car	0.91	0.765	0.84	0.659
	bus	0.881	0.949	0.958	0.869
	truck	0.891	0.564	0.669	0.544
	van	0.871	0.74	0.817	0.674
	walker	0.927	0.5	0.6	0.41
	traffic light	0.96	0.538	0.667	0.553
Fog density 25%	car	0.935	0.848	0.929	0.75
	bus	0.981	0.891	0.903	0.799
	truck	0.921	0.821	0.909	0.785
	van	0.932	0.878	0.949	0.831
	walker	0.941	0.639	0.778	0.485
	traffic light	0.976	0.92	0.965	0.818
Fog density 50%	car	0.944	0.687	0.79	0.637
	bus	0.853	0.871	0.9	0.779
	truck	0.843	0.612	0.669	0.555
	van	0.918	0.656	0.751	0.617
	walker	0.964	0.511	0.61	0.426
	traffic light	0.982	0.549	0.649	0.528
Fog density 75%	car	0.915	0.597	0.697	0.558
	bus	0.95	0.895	0.928	0.812
	truck	0.893	0.487	0.566	0.474
	van	0.917	0.577	0.672	0.558
	walker	0.94	0.419	0.506	0.354
	traffic light	0.974	0.528	0.616	0.494
Fog density 100%	car	0.905	0.589	0.677	0.55
	bus	0.926	0.629	0.693	0.57
	truck	0.907	0.424	0.497	0.425
	van	0.873	0.608	0.653	0.578
	walker	0.892	0.432	0.5	0.352
	traffic light	0.973	0.516	0.585	0.461

Table 8. We evaluated our models across all fog categories for each model. We can conclude from the table that the precision and recall are significantly higher when using a model trained on the same fog density. The labels in this validation are for objects within 100 m using YOLOv8m.

Metrics		Validation Data					
		Fog Density	0%	25%	50%	75%	100%
Trained model	Precision	0%	0.915	0.895	0.86	0.807	0.821
		25%	0.904	0.924	0.897	0.897	0.853
		50%	0.873	0.907	0.913	0.926	0.933
		75%	0.853	0.874	0.901	0.945	0.94
		100%	0.868	0.859	0.897	0.924	0.931
	Recall	0%	0.665	0.58	0.464	0.36	0.271
		25%	0.631	0.692	0.652	0.553	0.468
		50%	0.537	0.625	0.716	0.639	0.596
		75%	0.462	0.589	0.678	0.668	0.61
		100%	0.434	0.558	0.665	0.663	0.636

In general, it should be also noted that for autonomous driving vehicles, it is of crucial importance to detect correctly the state (red, yellow, green) of a traffic light. This will be a subject of further study.

5. Conclusions

Our primary objective in this study was to enhance the perception of traffic participants and traffic lights under dense fog conditions by developing models that are tailored to specific fog density levels. This approach allows our system to prioritize the relevant features of objects in fog, leading to improved detection accuracy. Furthermore, this approach enhances the flexibility of autonomous driving (AD) in severe weather conditions by enabling the use of specialized algorithms tailored to specific fog density categories. Additionally, it enables the detection of objects that are not visible to the human eye using only RGB images. This capability becomes even more efficient when combined with other sensors such as LiDAR and radar. As we observed, the core of the algorithm focuses on creating a separate model for each fog category (clear, low fog, moderate fog, etc.), which improves recall and precision compared to a model trained for general weather conditions (see Table 8).

For future research, we intend to extend our methodology to real-world data, aiming to improve object detection under actual environmental conditions. A primary challenge in utilizing real data will involve creating specialized datasets that categorize each level of fog density in addition to performing object detection.

This study demonstrates that classifying fog density enhances perceptual accuracy by increasing recall and precision. As illustrated in Table 7, classifying fog and training each model based on fog density yields improved precision. These findings underscore the importance of fog classification, particularly given the absence of existing datasets that categorize fog levels and provide labeled bounding boxes, which are notable challenges. This study thus highlights the critical role of fog classification.

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

Funding: This research received no external funding.

Data Availability Statement: All the data generated during our project is available on GitHub (<https://github.com/Mofeed-Chaar/Improving-bounding-box-in-Carla-simulator>, accessed on 2 October 2024). Additionally, we are happy to share the full dataset upon request via email.

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

Appendix A

The fog density refers to how much visibility is reduced by fog and other atmospheric particles. For instance, 100% obscuration implies very dense fog where visibility is minimal, while lower percentages represent lighter fog. The following table explains the relation between the fog density and visibility [44].

Table A1. Relation between fog density and visibility.

Visibility Distance		Fog Category	
min	max	kind	percentage
1000 m	∞ m	No Fog	0%
300 m	1000 m	Low Fog	25%
100 m	300 m	Moderate Fog	50%
50 m	100 m	Dense Fog	75%
0 m	50 m	Very Dense Fog	100%

Appendix B

Camera parameters [45]:

Table A2. Camera attributes.

Blueprint Attribute	Value	Description
bloom intensity	0.675	Intensity for the bloom
fov	90.0	Horizontal field of view
fstop	1.4	Opening of the camera lens. Aperture is $1/fstop$
image width	1280	in pixels
image height	720	in pixels
lens flare intensity	0.1	Intensity for the lens flare post-process effect.

Appendix C

Table A3. The hyperparameters that we used in YoloV5 and YoloV8.

Parameter Name	Value
epochs	50
batch	16
IOU	0.7
lr0	0.01
lrf	0.01
momentum	0.937
weight decay	0.0005
warmup momentum	0.8
warmup bias lr	0.1

References

1. Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access* **2020**, *8*, 58443–58469. [\[CrossRef\]](#)
2. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The kitti vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361. [\[CrossRef\]](#)
3. Oeljeklaus, M. An integrated approach for traffic scene understanding from monocular cameras. In *Eldorado-Repository*; Technical University of Dortmund: Dortmund, Germany, 2020. [\[CrossRef\]](#)
4. Sengupta, A.; Cheng, L.; Cao, S. Robust multiobject tracking using mmwave radar-camera sensor fusion. *IEEE Sens. Lett.* **2022**, *6*, 1–4. [\[CrossRef\]](#)
5. Lefsky, M.A.; Cohen, W.B.; Parker, G.G.; Harding, D.J. Lidar remote sensing for ecosystem studies: Lidar, an emerging remote sensing technology that directly measures the three-dimensional distribution of plant canopies, can accurately estimate vegetation structural attributes and should be of particular interest to forest, landscape, and global ecologists. *BioScience* **2002**, *52*, 19–30. [\[CrossRef\]](#)
6. Raiyn, J. Detection of road traffic anomalies based on computational data science. *Discov. Internet Things* **2022**, *2*, 6. [\[CrossRef\]](#)
7. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An open urban driving simulator. In Proceedings of the 1st Annual Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; Levine, S., Vanhoucke, V., Goldberg, K., Eds.; ML Research Press: Zurich, Switzerland, 2017. Available online: <http://proceedings.mlr.press/v78/dosovitskiy17a/dosovitskiy17a.pdf> (accessed on 2 October 2024).
8. Niranjana, D.; VinayKarthik, B.; Mohana. Deep learning based object detection model for autonomous driving research using carla simulator. In Proceedings of the 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 7–9 October 2021; pp. 1251–1258. [\[CrossRef\]](#)
9. Chaar, M.M.; Weidl, G.; Raiyn, J. *Analyse the Effect of Fog on the Perception*; EU Science Hub: Brussels, Belgium, 2023; p. 332. [\[CrossRef\]](#)
10. Zhang, Y.; Carballo, A.; Yang, H.; Takeda, K. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS J. Photogramm. Remote Sens.* **2023**, *196*, 146–177. [\[CrossRef\]](#)
11. Diaz-Ruiz, C.A.; Xia, Y.; You, Y.; Nino, J.; Chen, J.; Monica, J.; Chen, X.; Luo, K.; Wang, Y.; Emond, M.; et al. Ithaca365: Dataset and driving perception under repeated and challenging weather conditions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 21383–21392. [\[CrossRef\]](#)

12. Chaar, M.; Raiyn, J.; Weidl, G. Improve Bounding Box in Carla Simulator. In Proceedings of the 10th International Conference on Vehicle Technology and Intelligent Transport Systems-VEHITS, Angers, France, 2–4 May 2024; INSTICC, SciTePress: Setúbal, Portugal, 2024; pp. 267–275. [\[CrossRef\]](#)
13. Valanarasu, J.M.J.; Yasarla, R.; Patel, V.M. Transweather: Transformer-based restoration of images degraded by adverse weather conditions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 2353–2363. [\[CrossRef\]](#)
14. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Truong, T.N.; Nguyen, C.T.; Zanibbi, R.; Mouchère, H.; Nakagawa, M. A survey on handwritten mathematical expression recognition: The rise of encoder-decoder and GNN models. *Pattern Recognit.* **2024**, *153*, 110531. [\[CrossRef\]](#)
16. Bijelic, M.; Gruber, T.; Mannan, F.; Kraus, F.; Ritter, W.; Dietmayer, K.; Heide, F. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11682–11692. [\[CrossRef\]](#)
17. Li, J.; Xu, R.; Ma, J.; Zou, Q.; Ma, J.; Yu, H. Domain adaptive object detection for autonomous driving under foggy weather. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 2–7 January 2023; pp. 612–622. [\[CrossRef\]](#)
18. Ogunrinde, I.; Bernadin, S. A review of the impacts of defogging on deep learning-based object detectors in self-driving cars. In Proceedings of the SoutheastCon 2021, Atlanta, GA, USA, 10–13 March 2021; pp. 1–8. [\[CrossRef\]](#)
19. Reis, D.; Kupec, J.; Hong, J.; Daoudi, A. Real-Time Flying Object Detection with YOLOv8. *arXiv* **2023**, arXiv:2305.09972.
20. Farhadi, A.; Redmon, J. Yolov3: An incremental improvement. In Proceedings of the Computer Vision and Pattern Recognition; Springer: Berlin/Heidelberg, Germany, 2018; Volume 1804, pp. 1–6.
21. Mallett, A.; Mallett, A. Writing YAML and Basic Playbooks. *Red Hat Certified Engineer (RHCE) Study Guide: Ansible Automation for the Red Hat Enterprise Linux 8 Exam (EX294)*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 63–77. [\[CrossRef\]](#)
22. Waymo. Waymo Dataset. 2023. Available online: <https://waymo.com/open/> (accessed on 6 December 2023).
23. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223. [\[CrossRef\]](#)
24. Cordts, M.; Omran, M.; Ramos, S.; Scharwächter, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset. In Proceedings of the CVPR Workshop on the Future of Datasets in Vision, Boston, MA, USA, 7–12 June 2015; Volume 2.
25. Heinzler, R.; Schindler, P.; Seekircher, J.; Ritter, W.; Stork, W. Weather influence and classification with automotive lidar sensors. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 1527–1534. [\[CrossRef\]](#)
26. TNO. Integrated Vehicle Safety—Autonomous Emergency Braking (AEB). 2015. Available online: https://www.youtube.com/watch?v=yNRgrOI329I&ab_channel=TNO (accessed on 6 December 2023).
27. Sakaridis, C.; Dai, D.; Van Gool, L. Semantic foggy scene understanding with synthetic data. *Int. J. Comput. Vis.* **2018**, *126*, 973–992. [\[CrossRef\]](#)
28. Muller, R. Drivetruth: Automated autonomous driving dataset generation for security applications. In Proceedings of the Workshop on Automotive and Autonomous Vehicle Security (AutoSec), San Diego, CA, USA, 24 April 2022. [\[CrossRef\]](#)
29. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional neural networks: An overview and application in radiology. *Insights Imaging* **2018**, *9*, 611–629. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Raiyn, J.; Weidl, G. Naturalistic Driving Studies Data Analysis Based on a Convolutional Neural Network. In Proceedings of VEHITS 2023: 9th International Conference on Vehicle Technology and Intelligent Transport Systems, Prague, Czech Republic, 26–28 April 2023. [\[CrossRef\]](#)
31. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [\[CrossRef\]](#)
32. Sarda, A.; Dixit, S.; Bhan, A. Object detection for autonomous driving using yolo [you only look once] algorithm. In Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 4–6 February 2021; pp. 1370–1374. [\[CrossRef\]](#)
33. Zhang, Y.; Guo, Z.; Wu, J.; Tian, Y.; Tang, H.; Guo, X. Real-time vehicle detection based on improved yolo v5. *Sustainability* **2022**, *14*, 12274. [\[CrossRef\]](#)
34. Powers, D.M. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv* **2020**, arXiv:2010.16061.
35. Juba, B.; Le, H.S. Precision-recall versus accuracy and the role of large data sets. *Proc. Aaai Conf. Artif. Intell.* **2019**, *33*, 4039–4048. [\[CrossRef\]](#)
36. Chicco, D.; Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genom.* **2020**, *21*, 6. [\[CrossRef\]](#)
37. Pereira, N. PereiraASLNet: ASL letter recognition with YOLOX taking Mean Average Precision and Inference Time considerations. In Proceedings of the 2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP), Vijayawada, India, 12–14 February 2022; pp. 1–6. [\[CrossRef\]](#)

38. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An Open Urban Driving Simulator. 2017. Available online: https://carla.readthedocs.io/en/latest/python_api/#carlaweatherparameters (accessed on 24 December 2023).
39. Fu, Q.; Luo, K.; Song, Y.; Zhang, M.; Zhang, S.; Zhan, J.; Duan, J.; Li, Y. Study of sea fog environment polarization transmission characteristics. *Appl. Sci.* **2022**, *12*, 8892. [[CrossRef](#)]
40. Ivanov, H.; Leitgeb, E. Artificial Generation of Mie Scattering Conditions for FSO Fog Chambers. In Proceedings of the 2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), Porto, Portugal, 20–22 July 2022; pp. 54–58. [[CrossRef](#)]
41. Haider, A.; Pigniczki, M.; Koyama, S.; Köhler, M.H.; Haas, L.; Fink, M.; Schardt, M.; Nagase, K.; Zeh, T.; Eryildirim, A.; et al. A Methodology to Model the Rain and Fog Effect on the Performance of Automotive LiDAR Sensors. *Sensors* **2023**, *23*, 6891. [[CrossRef](#)] [[PubMed](#)]
42. Ultralytics Inc. YoloV8. 2023. Available online: <https://docs.ultralytics.com/models/yolov8/> (accessed on 4 November 2024).
43. Liu, Y.; Gao, Y.; Yin, W. An improved analysis of stochastic gradient descent with momentum. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 18261–18271. [[CrossRef](#)]
44. Negru, M.; Nedeveschi, S. Assisting navigation in homogenous fog. In Proceedings of the 2014 International Conference on Computer Vision Theory and Applications (VISAPP), Lisbon, Portugal, 5–8 January 2014; Volume 2, pp. 619–626.
45. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An Open Urban Driving Simulator. 2017. Available online: https://carla.readthedocs.io/en/latest/ref_sensors/#rgb-camera (accessed on 4 November 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.