

Article

Object Detection Performance Evaluation for Autonomous Vehicles in Sandy Weather Environments

Nasser Aloufi ^{*,†}, Abdulaziz Alnori [†] , Vijey Thayanathan [†] and Abdullah Basuhail [†]

Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; asalnori@kau.edu.sa (A.A.); thayanathan@live.co.uk (V.T.); abasuhail@kau.edu.sa (A.B.)

* Correspondence: nhassanaloufi@stu.kau.edu.sa

[†] These authors contributed equally to this work.

Abstract: In order to reach the highest level of automation, autonomous vehicles (AVs) are required to be aware of surrounding objects and detect them even in adverse weather. Detecting objects is very challenging in sandy weather due to characteristics of the environment, such as low visibility, occlusion, and changes in lighting. In this paper, we considered the You Only Look Once (YOLO) version 5 and version 7 architectures to evaluate the performance of different activation functions in sandy weather. In our experiments, we targeted three activation functions: Sigmoid Linear Unit (SiLU), Rectified Linear Unit (ReLU), and Leaky Rectified Linear Unit (LeakyReLU). The metrics used to evaluate their performance were precision, recall, and mean average precision (mAP). We used the Detection in Adverse Weather Nature (DAWN) dataset which contains various weather conditions, though we selected sandy images only. Moreover, we extended the DAWN dataset and created an augmented version of the dataset using several augmentation techniques, such as blur, saturation, brightness, darkness, noise, exposer, hue, and grayscale. Our results show that in the original DAWN dataset, YOLOv5 with the LeakyReLU activation function surpassed other architectures with respect to the reported research results in sandy weather and achieved 88% mAP. For the augmented DAWN dataset that we developed, YOLOv7 with SiLU achieved 94% mAP.

Keywords: autonomous vehicles; conventional neural network; object detection; deep learning; sandy weather



Citation: Aloufi, N.; Alnori, A.; Thayanathan, V.; Basuhail, A. Object Detection Performance Evaluation for Autonomous Vehicles in Sandy Weather Environments. *Appl. Sci.* **2023**, *13*, 10249. <https://doi.org/10.3390/app131810249>

Academic Editor: DaeEun Kim

Received: 10 August 2023

Revised: 10 September 2023

Accepted: 11 September 2023

Published: 13 September 2023



Copyright: © 2023 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

Recognizing objects in the surrounding area is a crucial milestone in the development of autonomous vehicles (AVs). To eliminate the risk of accidents and to navigate safely in various environmental conditions, AVs are required to scan the scene through their equipped sensors, localize objects, classify objects, and finally, take action (in the form of a response) based on the detected data that they have collected.

The above process is very challenging when it comes to detection in adverse weather conditions such as rain, fog, and sand. Each of these weather conditions presents unique challenges that can affect the performance of object detection systems. Sand, rain, and snow can obscure objects, reduce visibility, and create reflections on surfaces, making it difficult for sensors to accurately detect and recognize objects. Fog can significantly limit visibility and cause signal attenuation, further degrading the accuracy of object detection algorithms. Harsh sunlight can lead to glare and cast strong shadows, making it challenging to distinguish objects from their surroundings.

One significant milestone in the development of AVs and object detection is the transition from the traditional computer vision methods to deep learning-based approaches, particularly with the use of Convolutional Neural Networks (CNNs). This milestone represents a shift towards more powerful and accurate object detection models in various scenarios and weather conditions. The milestone was marked by the emergence of

landmark algorithms such as Regions with Convolutional Neural Networks (R-CNNs) [1] and its subsequent variants, including Fast Region-based Convolutional Network (Fast R-CNN) and Faster Region-based Convolutional Network (Faster R-CNN) [2,3]. These algorithms, which are called two stage models, introduced the concept of region proposal methods, in which potential object regions are first identified and then classified using CNNs. This breakthrough significantly improved object detection accuracy in various weather conditions and laid the foundation for subsequent advancements. Another significant milestone was reached with the development of the You Only Look Once (YOLO) algorithm. These milestones introduced the concept of real-time object detection performed directly on a single network pass (one stage), eliminating the need for region proposal methods. Additionally, they marked a significant improvement in computational efficiency while maintaining competitive detection performance. These advancements in object detection have paved the way for more recent milestones, such as the appearance of anchor-free approaches (e.g., CenterNet and EfficientDet [4,5]) and the incorporation of transformers (e.g., DETR [6]). Through these milestones, researchers continue to push the boundaries of object detection, improving accuracy, speed, and robustness for a wide range of applications in various domains.

The growth of sandstorms [7,8] reflects the importance of developing robust object detection algorithms and sensor technologies that can adapt to the challenges of adverse weather conditions. Dense airborne particles could be misinterpreted as an object ahead of the vehicle and possibly result in a false positive action from the AV [9]. Drastic changes in lighting with sudden darkness during sandstorms may lead to poor image quality and make it challenging for the vehicle's perception system to accurately recognize and track objects. Road collisions with animals are also a serious issue in sandy environments, such as the Middle East, where animals, specifically camels, can often be seen walking alongside or crossing roads, causing major problems for road safety. The authorities in Middle East countries are continuously working to find solutions to this problem, but it remains a significant challenge for road safety and AVs. From a computing standpoint, great efforts are being made to provide road animal datasets and to apply detection in this domain, as in [10,11]. However, we are still lacking a reliable dataset that could enhance research in this domain.

In fact, there are several key obstacles that AVs may face in sandy weather environments:

- As a common occurrence in deserts and coastal regions, dust and sand particles in the air can severely impair visibility and reduce the accuracy of object detection algorithms.
- Occlusion is where objects are covered by other objects, making perception very difficult for the deep learning model. Occlusion makes it difficult for detection algorithms to correctly determine object boundaries and characteristics.
- Changes in lighting during storms can affect the performance of cameras and sensors used in object detection.
- Road collisions with wildlife animals are a significant risk for vehicles, and it has been reported that wildlife animals are causing risk for vehicles [12].
- There is a lack of sandy weather datasets, with most of the public datasets focusing on other types of weathers (foggy, snowy, and rainy).

Figure 1 shows some object detection challenges in sandy weather. We can clearly see that sandstorms and sandy environments can drastically change scene lighting and reduce visibility, making it difficult for sensors to accurately perceive objects.



Figure 1. Some sandy weather challenges. Figures (a,b) demonstrate changes in lighting during a sandstorm. Figures (c,d) illustrate the degradation of visibility during a sandstorm. In figures (e,f), the road boundaries are obscured by sand.

All of the previously mentioned limitations are challenges in sandy weather. AVs heavily rely on object detection systems to perceive and interpret their environment. The challenges posed by sandy weather can significantly impact their safety and reliability, which are two critical aspects in the development and deployment of AVs. Overcoming these challenges will play a vital part in the transition toward a future with safer and more reliable autonomous transportation systems.

In this paper, we tackle two of the challenges that arise in sandy weather. The first is the challenge of lacking datasets by providing an augmented dataset, and the second is the visibility of the scene and to evaluate the performance of YOLOv5 and YOLOv7 in such weather. Metrics such as precision, recall, and mAP will be our bases for evaluation. Figure 2 shows the scope of our paper which focuses on detecting objects using the camera (as a sensor) in sandy weather.

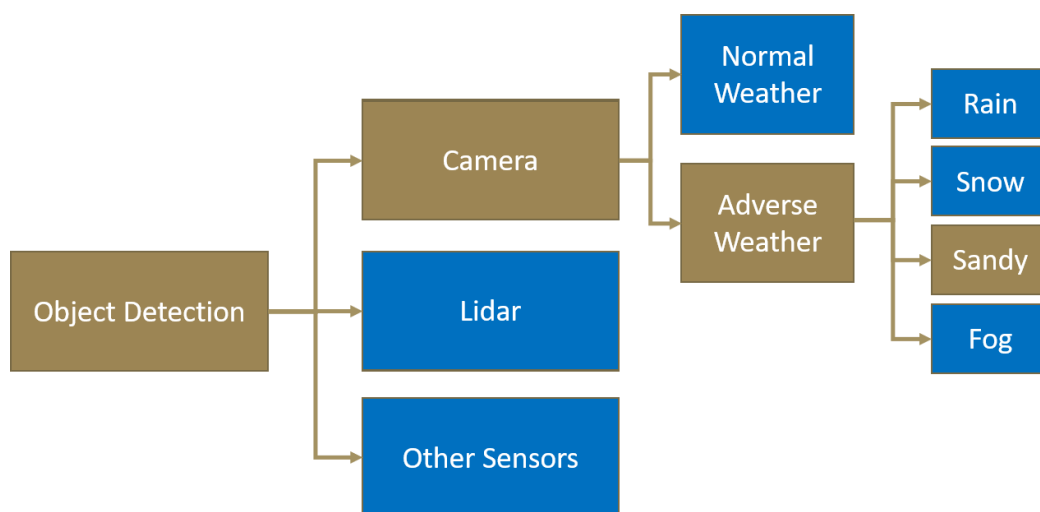


Figure 2. Scope of our paper.

The main contributions of our work are as follows:

- We extended the Detection in Adverse Weather Nature (DAWN) dataset and added augmented images. The sandy weather dataset was expanded from 323 images to 1137 images. The augmentations that were used include saturation, brightness, darkness, blur, noise, exposure, hue, and gray scale.
- We used object detection models (YOLOv5 and YOLOv7) as base architectures for detecting three classes of objects (car/vehicle, person, and bicycle) in sandy weather.
- We evaluated different activation functions for detecting objects in sandy weather.

The following section will shed light on activation functions and recent related work on object detection in adverse weather conditions. In the next section, we will provide a detailed explanation of the dataset, methodology, and evaluation metrics used in our experiments. Subsequently, we will present the results of the experiments conducted using the original DAWN dataset, followed by those obtained using the augmented DAWN dataset. The final part of this paper will focus on discussing the outcomes of the experiments and their evaluation, before concisely summarizing the findings.

2. Background and Related Work

We divided our discussion on the study background and related work into three parts. The first part concerns the most popular activation functions that have been used in object detection models. The second part concerns the most popular object detection models. The third part presents the related works on object detection in adverse weather.

2.1. Activation Functions

Activation functions are crucial parameters of neural networks in deep learning. These functions introduce non-linearity into the network by determining the output of a neuron. Without an activation function, the network would not be able to learn new features and would be limited to a linear mapping of the input. The activation function acts as a gatekeeper by deciding whether a neuron should be activated or not based on the input signal it receives. By transforming the input signal and introducing non-linearity, the activation function allows the network to model complex relationships and capture intricate patterns in the data. There are various activation functions used in deep learning, each with its own advantages and limitations. The following subsections outline three of the most used activation functions in deep learning.

2.1.1. Rectified Linear Unit (ReLU) Function

ReLU is one of the most widely used functions in deep learning. ReLU was introduced in the year 2010 [13] and over the past few years has proven its value as a function that improves the deep learning process [14,15]. The authors of [16] used ReLU for training a deep neural network and achieved results faster than when using logistic units. The idea behind ReLU's fast computation is that it counts any negative result as zero without any multiplication or division operation. New variations of ReLU have been introduced such as LeakyReLU [17] and Parametric Rectified Linear Unit (PReLU) [18].

2.1.2. Leaky Rectified Linear Unit (LeakyReLU) Function

LeakyReLU is a variant of the ReLU activation function that solves the dying ReLU problem. The LeakyReLU function introduces a small positive slope (typically 0.01) to negative values instead of forcing them to zero, as in the original ReLU function. This slight slope prevents neurons from dying and encourages a more robust gradient flow, which improves the performance of deep neural networks. LeakyReLU has been used in various deep learning models and has been shown to outperform other activation functions like sigmoid and tanh. In fact, it has become a standard activation function in many popular deep learning libraries like TensorFlow and PyTorch. Moreover, different variants of LeakyReLU have been proposed to improve its performance, such as Parametric Rectified Linear Unit (PReLU), which allows the slope of the negative region to be learned during

training, and Exponential Linear Unit (ELU), which introduces an exponential term for negative inputs. The LeakyReLU function has become a popular choice for deep learning applications due to its ability to overcome the limitations of the original ReLU function and its effectiveness in improving the training process of deep neural networks.

2.1.3. Sigmoid Linear Unit (SiLU) Function

Compared to Sigmoid and ReLU, SiLU (also called Swish) is relatively new. It was proposed in 2017 by the authors of [19] and can serve as a replacement function for ReLU. The Swish function has been used in the proposed YOLOv5 model and achieved remarkable results.

2.2. Object Detection

Detection objects using CNNs can be divided into two approaches: the one-stage approach and two-stage approach. The two-stage approach was initially introduced in 2013 when the Region-Based CNN (R-CNN) model was developed. The first stage of the model is called region proposal, which aims to find the regions in the image that contain objects. R-CNN generates approximately 2000 region proposals. The second stage in the CNN is generally the extraction of features from the proposed regions followed by object classification. Because with every proposed region a different network is used, R-CNN remained slow. A new version of R-CNN was introduced in the following year named Fast R-CNN, which aims to address the slowness of R-CNN. Fast R-CNN does not pass 2000 proposals to the CNN; instead, it passes the entire image, and then a feature map is generated, which significantly improves the speed of detection. Another version of the R-CNN family called Faster R-CNN followed. In 2017, a remarkable improvement was achieved when Mask R-CNN was introduced [20]. Mask R-CNN uses Feature Pyramid Network (FPN) [21] as its backbone and adds a new phase to the process of detection which is a segmentation mask for every object.

The one-stage object detection approach was first introduced by Redmon et al. [22] with the YOLO model. The whole process of detection is encapsulated within a single pass to the CNN network. This model was a remarkable milestone for object detection since it provides both high-speed detection for high-FPS scenes (suitable for real-time detection) and good accuracy. After the first YOLO, a new model (YOLOv2) followed, which was trained on PASCAL Visual Object Classes (PASCAL VOCs) and Common Objects in Context (COCO) datasets and achieved a mAP of 76.8 on VOC 2007. These incremental improvements continued with YOLOv3 where a new backbone called Darknet-53 was proposed [23]. YOLOv4 was developed to improve the average precision and frames per second by approximately 10% and 12%, respectively, compared to the previous YOLO model [24]. YOLOv5 and then YOLOv7 in 2022 [25] followed. The authors of [26] proposed a new one-stage approach called Single-Shot multibox Detector (SSD). When SSD received an image input sized 300×300 , it achieved a mAP of approximately 74.3% on the VOC 2007 dataset. This score improved to 76.9% when the authors increased the size of the image to 512×512 .

2.3. Related Work

Object detection in adverse weather conditions is a challenging task due to the degradation of the image quality and the loss of visual features caused by weather phenomena such as rain, fog, snow, and haze. These weather conditions affect the performance of detection by reducing the contrast in the scene, decreasing the visibility of the object, and making it difficult to distinguish between objects and the surrounding elements. In [27], the authors used YOLOv4 with spatial pyramid pooling (SPP-NET) layers to detect cars in severe weather, including haze, dust, snow, and rain. Several augmentation techniques have been used to maximize the DAWN dataset such as hue, saturation, exposure, brightness, darkness, blur, and noise. The authors achieved an mAP of 81%. Unfortunately, this paper focused on detecting only one kind of object, which was car class. Despite applying

augmentation techniques, only two types of augmentations (hue and saturation) were added in sandy weather. In [28], the authors also aimed to detect vehicles in severe weather. YOLOv4 was proposed, with an anchor-free and decoupled head. The authors achieved an mAP of 60.3%. Similar to the previous paper, only one class was used, which would limit the model's performance in reality. The authors of [29] proposed a model for detecting and extracting high-precision vehicle motion data under various weather conditions. The proposed tracking model, called SORT++, provides data such as the vehicle trajectory, vehicle speed, and vehicle yaw angle. Although that paper presented a new dataset called Multi-Weather Vehicle Detection (MWVD), this dataset, like many others, lacks sandy weather. Only rainy and snowy are covered, with a lack of other global weathers. Another paper [30] introduced an image enhancement framework called Image-Adaptive YOLO (IA-YOLO). The input image passes through a pre-processing phase in the Differentiable Image Processing (DIP) filter and is then fed into the YOLOv3 model. Through this framework, every targeted image can be enhanced for better quality and subsequent accurate object detection performance. This framework is promising; however, more challenging weather conditions and scenarios need to be evaluated, instead of using only foggy images. The Dual-Subnet Network (DSNet) was introduced in Huang et al.'s [31] paper. The developed network consists of two sub-networks, which are the detection subnet and restoration subnet. This paper focused on foggy weather, and the authors achieved an approximately 51% mAP on their composed dataset and approximately 42% on the Foggy Driving dataset [32]. In [33], the authors collected a Street-Level Video dataset that provided 11 classes including pedestrians, vehicles, and traffic lights. YOLOv5 was used, and an mAP of 72.3% was achieved for detecting vehicles. Unfortunately, one kind of adverse weather was covered, specifically rain, while other challenging weather conditions were out of the paper's scope. In [34], the authors addressed challenging weather using images that were taken with drones. Considering the previously mentioned related works, we can clearly see a gap and a bias of research papers against sandy weather. From our point of view, one obvious reason for not including sandy weather in object detection experiments is the lack of sandy datasets, which we aimed to overcome by creating an augmented DAWN dataset. Table 1 shows a summary of recent adverse weather publications.

Table 1. Summary of recent object detection publications in various weather conditions.

Reference	Dataset	Augmentation	Number of Classes	Model
[27]	DAWN	Yes	1	YOLOv4
[28]	BDD-IW	Yes	1	YOLOv4
[29]	MWVD	No	1	YOLOv5
[30]	VOC RTTS	Yes	5	YOLOv3
[31]	Foggy	No	1	DSNet
[33]	Collected dataset	No	11	YOLOv5
[35]	Open Image	No	4	YOLOv4 Faster
[36]	COCO BDD100K	No	All COCO	YOLOv4 DSSD
[37]	KITTI	No	3	Tiny YOLO Complex YOLO
Ours	DAWN	No	6	YOLO5
	Aug. DAWN	Yes	3	YOLOv7

Table 2 shows a sample of recent publications on object detection and the kinds of weather they address. From the samples mentioned in the table, we can clearly see that most of the publications focus on foggy and rainy weathers, omitting other kinds of weather

conditions such as sandy conditions. This omission has caused a bias in object detection model evaluations and results since the results are always impacted by the training dataset.

Table 2. Types of adverse weather covered in recent object detection publications.

Ref.	Sandy	Foggy	Snowy	Rainy
[28]	×	✓	✓	✓
[29]	×	×	✓	✓
[30]	×	✓	×	×
[31]	×	✓	×	×
[33]	×	×	×	✓
[34]	×	×	✓	✓
[35]	×	✓	✓	✓

We can summarize the reasons for excluding sandy environments from object detection publications in two points:

- Limited resources: there is an absence of reliable datasets accurately depicting sandy conditions.
- Regional and weather bias: the regions where researchers and institutions are located, as well as the weather conditions in those regions, can inadvertently influence the scope of object detection studies.

Our research aimed to fill the gap by shedding light on the performance of one-stage models (specifically, YOLO versions) in sandy weather environments.

3. Methodology

Our aim was to evaluate and enhance the performance of object detection in sandy weather. In this section, we will explain in detail the dataset and methods that were used for our experiments. Figure 3 shows a block diagram of our methodology.

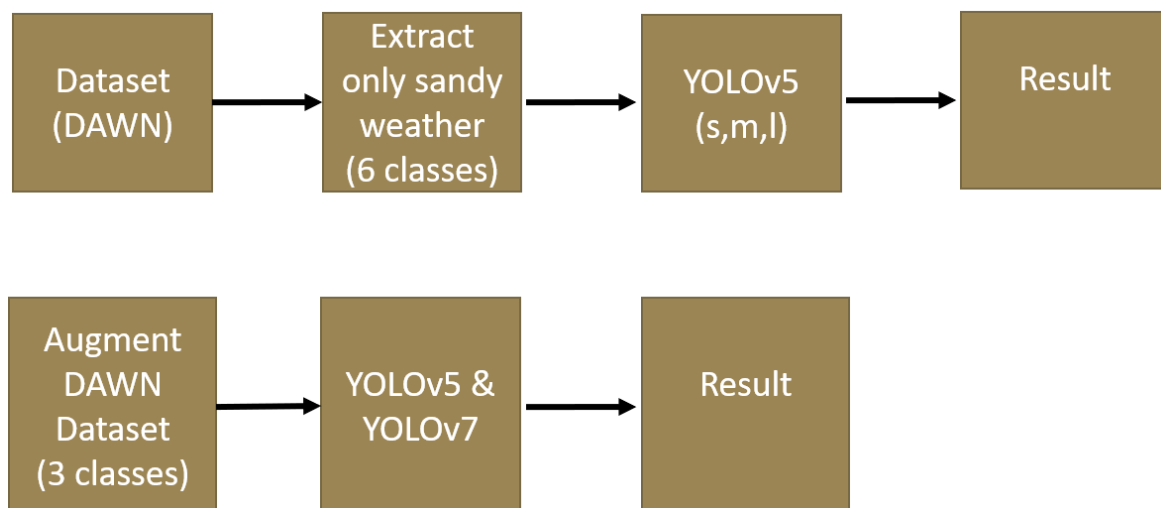


Figure 3. Sequence of our methodology in this paper.

Our dataset covers challenging sandy weather conditions. Detection in Adverse Weather Nature (DAWN), which was created by the authors of [38], was used as our main source of images. This dataset consists of approximately 1000 images that cover hard weather conditions, such as fog, snow, rain, and sandstorms, see Figure 4. It also covers various types of roads (urban, highways, and freeways). Desert countries have always been subjected to sandstorms, which represent a challenging type of weather for autonomous

vehicles and detection. Due to the fact that none of the previous publications focused on sandy weather, we decided to extract only sandy images from the DAWN dataset as our raw images. Our raw DAWN dataset was split into 258 images (80%) for training and 65 images (20%) for validation, which resulted in a total of 323 images sized 640×640 . We also narrowed our raw experiments to six objects: person, bicycle, car, motorcycle, bus, and truck. Image annotation was provided in the original dataset which contained the class of the object, the corresponding boundaries of x and y, and the width and height of the bounding box (x_center, y_center, width, and height). Figure 5 represents a sample of our labeled images considered as a ground truth reference.

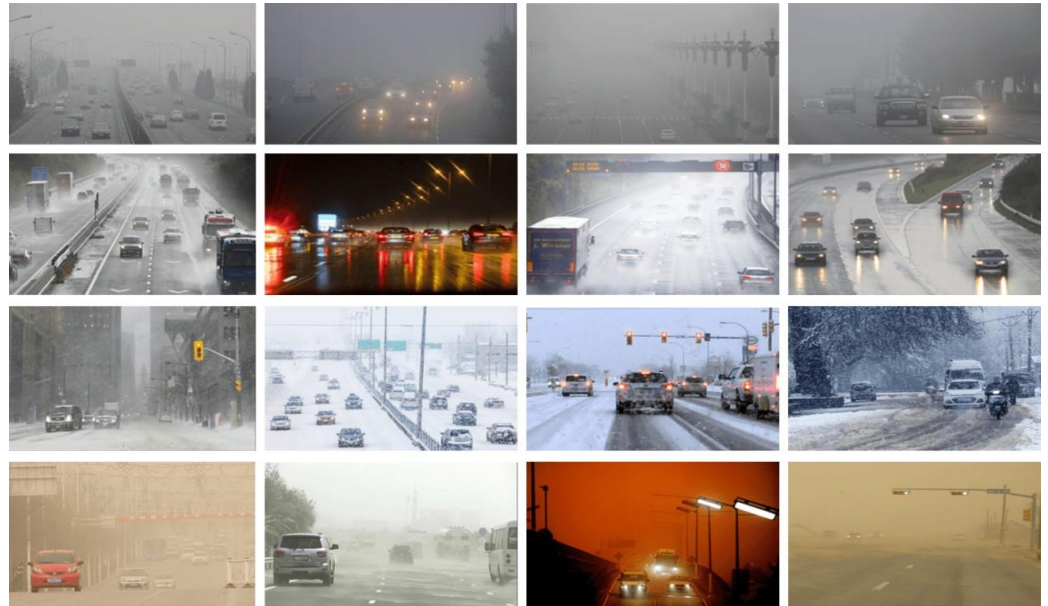


Figure 4. The DWAN dataset provides various hard weather conditions such as fog, rain, snow, and sand [38].



Figure 5. Sample of labeled images.

Due to the limited GPU resources available and to run our experiments as smoothly as possible, we used the Google cloud-based notebook “Colab” as a hosting environment for our experiments, using the machine learning framework “Pytorch”, with high Graphics Processing Unit (GPU) capabilities. Colab provides a high-performance GPU, such as Tesla T4 for public use. Colab also provides the Compute Unified Device Architecture (CUDA) which speeds up the computation of the CNN process, including convolution, pooling, normalization, and activating layers.

Several metrics can be used to measure the performance of the object detection model. In our research we focused on three main metrics: mean average precision (mAP), precision, and recall. mAP is the most commonly used evaluation metric for object detection. It provides a comprehensive measure of the model’s accuracy in terms of identifying and localizing objects. It combines precision and recall by calculating the average precision (AP) for each class or category of objects and then taking the mean across all classes. AP measures the quality of object detection by considering both the precision of correctly identified objects and the completeness of the detection. By computing the mAP, we can effectively compare different models and assess their performance in various domains.

Precision and recall are also other mandatory metrics for object detection. Precision will show the percentage of retrieved elements that are relevant, while recall is concerned with how many relevant items have been retrieved.

Precision is the ratio of TP to TP + FP:

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

Recall is the ratio of TP to TP + FN:

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

Additionally, as one of our metrics, we considered the F1 score, which combines precision and recall into a single value, providing a balanced measure of the model’s performance.

4. Original DAWN and YOLOv5 Results

After running our experiments with four batches for 32 epochs, SiLU generally outperformed all the other activation functions among the YOLO5s and scored the highest mAP at 80%, approximately 5 percentage points higher than the closest activation function in YOLO5s. Figure 6 shows the detected objects with their scores using the SiLU function. Although SiLU scored the highest mAP among the YOLOv5s, if we observe Figure 6, the hard sandstorm represented in the bottom left scene and the distant location of the objects prevented the model from detecting all the people in the scene. The model failed to detect the second person. Failure to detect all the elements in a scene is considered a high risk in real-life situations. The detailed SiLU metrics are shown in Figure 7, after running the model for 32 epochs. The top left chart shows the precision result, with a score of 79%, while the top right chart shows the recall result reaching 76%. The bottom chart shows the resulting mAP which reached 80%.

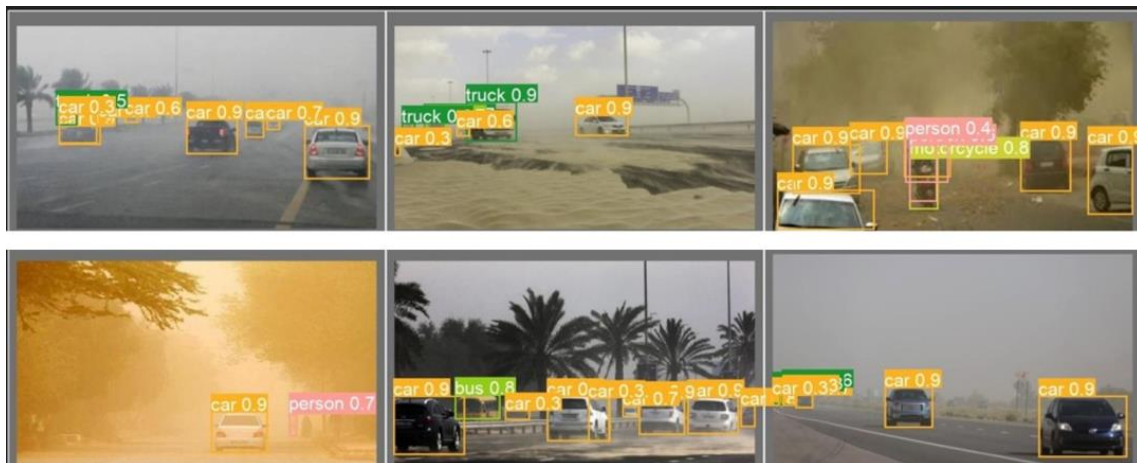


Figure 6. YOLO5s model predicted objects using SiLU.

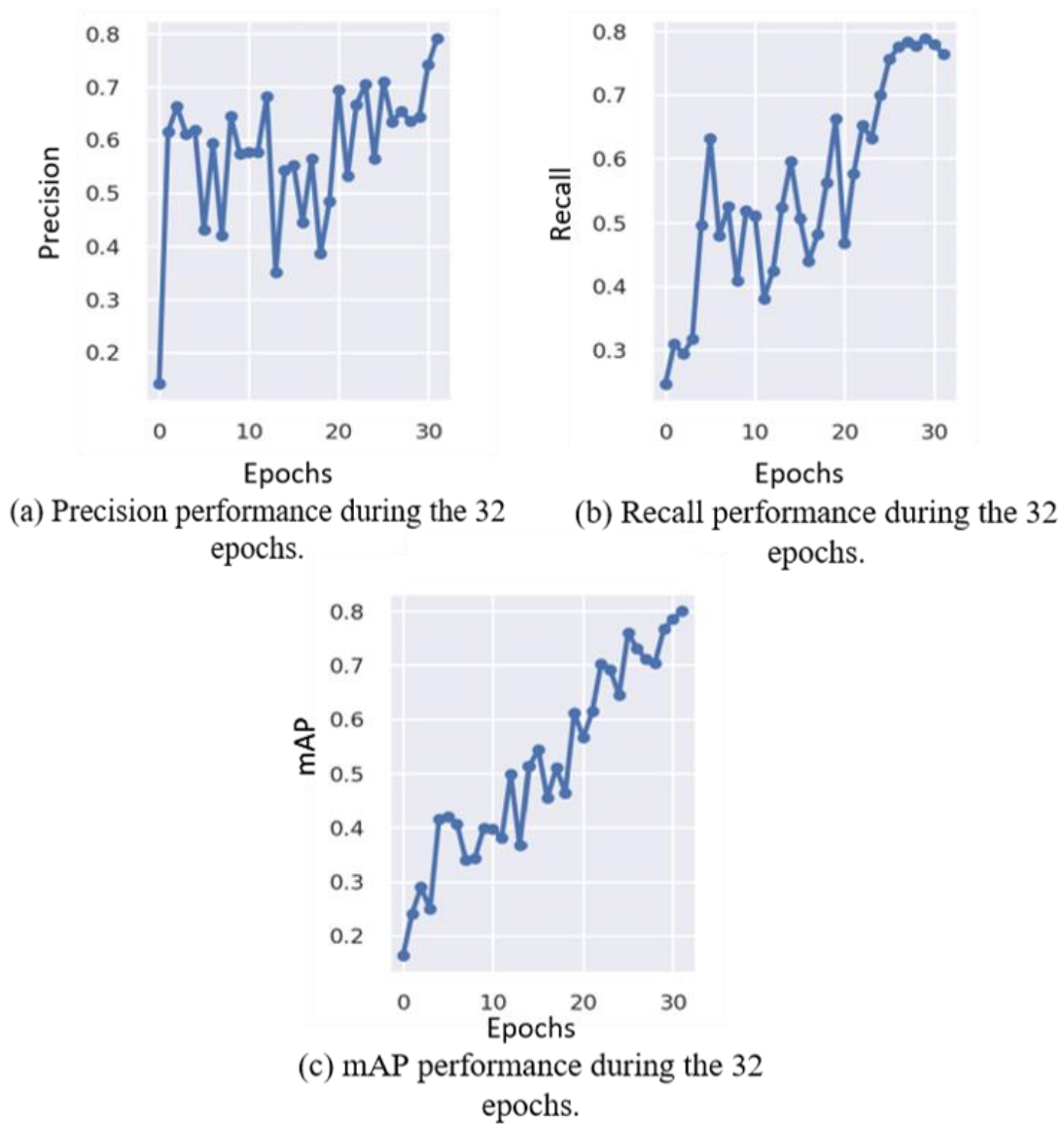


Figure 7. SiLU result after 32 epochs with YOLO5s model.

In the case of YOLO5m, the results were different. LeakyReLU achieved the highest mAP and highest precision. ReLU had a slightly better performance in terms of detecting the relevant objects in the scene (recall), since it achieved 70%, whereas SiLU and LeakyReLU scored 69% and 12%, respectively. Table 3 shows a summary of the YOLOv5 experimental results.

Table 3. Summary of results using original DAWN dataset and YOLOv5.

Model	Function	Image Size	mAP	Precision	Recall
YOLO5s	SiLU	640	80%	79%	76%
YOLO5s	ReLU	640	75%	59%	55%
YOLO5s	LeakyReLU	640	71%	10%	11%
YOLO5m	SiLU	640	85%	73%	69%
YOLO5m	ReLU	640	82%	80%	70%
YOLO5m	LeakyReLU	640	88%	97%	12%

Figure 8 shows part of the results when we detected objects using YOLOv5s version 5 on the testing video frames with SiLU as an activation function. From the figure, we can see that the detection score for the black vehicle on the right reached 92%. However, if we observe the other side of the road, the YOLOv5s with SiLU could not detect objects in the far distance. The model did not detect the white truck on the other side.

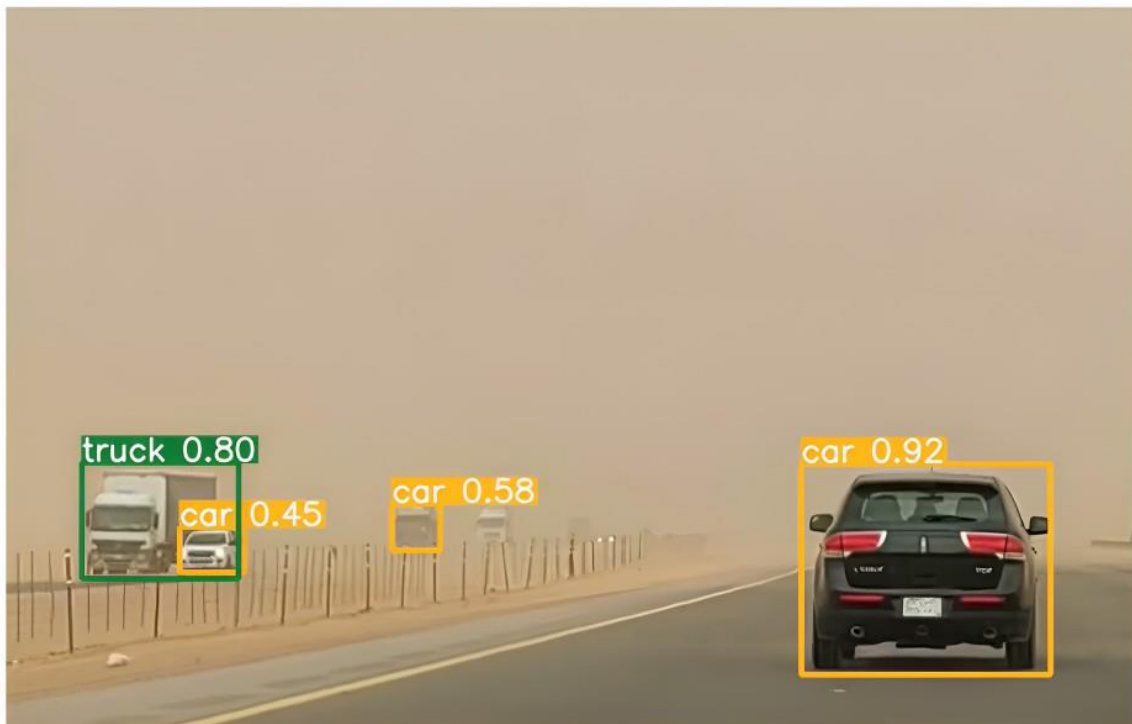


Figure 8. Detection result for YOLOv5s.

5. Augmented DAWN Dataset

Because the DAWN dataset has a limited number of sandy weather images, we expanded the dataset by creating new augmented images. Data augmentation is a common technique that artificially expands the dataset by generating additional training samples from the existing data. Several researchers have used data augmentation for object detection, e.g., refs. [39,40]. Data augmentation can involve various operations, including image scaling, rotation, cropping, flipping, colorization, and the addition of noise or blur.

Additionally, object-specific augmentations can be performed by applying geometric transformations, such as scaling, rotation, and translation, to the bounding boxes that enclose the objects of interest. Augmenting the object detection dataset can benefit the model in several ways:

1. Increasing the diversity and variability of the training data, which can help to generalize the model to unrepresented scenarios.
2. Improving the model's robustness against various factors that may affect the object's appearance or shape, such as different lighting conditions, occlusions, or viewpoint changes.
3. Balancing the class distribution in the dataset by oversampling the minority classes or undersampling the majority ones.
4. Reducing overfitting by introducing regularization and noise to the training data.

For the DAWN dataset, we performed several augmentations to expand our dataset. The following are the augmentations we performed:

5.0.1. Blur

Blur is used to introduce out-of-focus effects into images. For our augmented data, we used Gaussian blur with up to 1.25 px.

5.0.2. Saturation

Saturation alters the intensity of colors in the image. When we saturate an image, we basically multiply the pixel values by a random factor within a certain range. Increasing the saturation value of an image can make the colors more vibrant and vivid, while decreasing it can make the colors more subdued and muted. We augmented the saturation of our dataset by approximately 50%.

5.0.3. Brightness

By randomly increasing the brightness of an image, we exposed our model to a wider range of lighting conditions, making it more robust to changes in illumination. We augmented the images and made them approximately 20% brighter.

5.0.4. Darkness

This is the opposite of the previous operation, where darkness applied to an image. This augmentation can be helpful for simulating scenarios where the lighting conditions are poor, as in the night time, or in bright lighting conditions, such as bright sunlight.

5.0.5. Noise

We also added synthetic noise to maximize our dataset. This kind of augmentation makes the model more robust to noise and improves its ability to handle new data or scenarios.

5.0.6. Exposure

Additionally, we artificially modified the exposure level of the images, setting it in the range of 15% to −15%.

5.0.7. Hue

Our augmentations also included hue, which is a color-based image augmentation technique that changes the hue or color tone of an image without affecting its brightness or saturation.

5.0.8. Grayscale

Finally, we added grayscale augmentation that converts an image into grayscale. This technique is commonly used to increase the contrast of an image and enhance its details.

Table 4 shows our augmentation setting values and their impacts on images. After performing the augmentations, the number of DAWN sandy images increased from only

323 images to 1137 images. Figure 9 shows a general view of our augmentation techniques. Figures 10–16 show samples of our conducted augmentations.

Table 4. Summary of applied augmentations and their impact on images.

Augmentation	Value	Impact
Blur	1.25 px	Averaging pixel values within neighboring ones.
Saturation	50%	Changes the intensity of pixels.
Brightness	20%	Image appears lighter.
Darkness	20%	Image appears darker.
Noise	Random noise added	More obstacles added to the image.
Exposure	15%	More resilient to lighting and camera setting changes.
Hue	90%	Random adjustment of colors.
Grayscale	25%	Converts image to single channel.

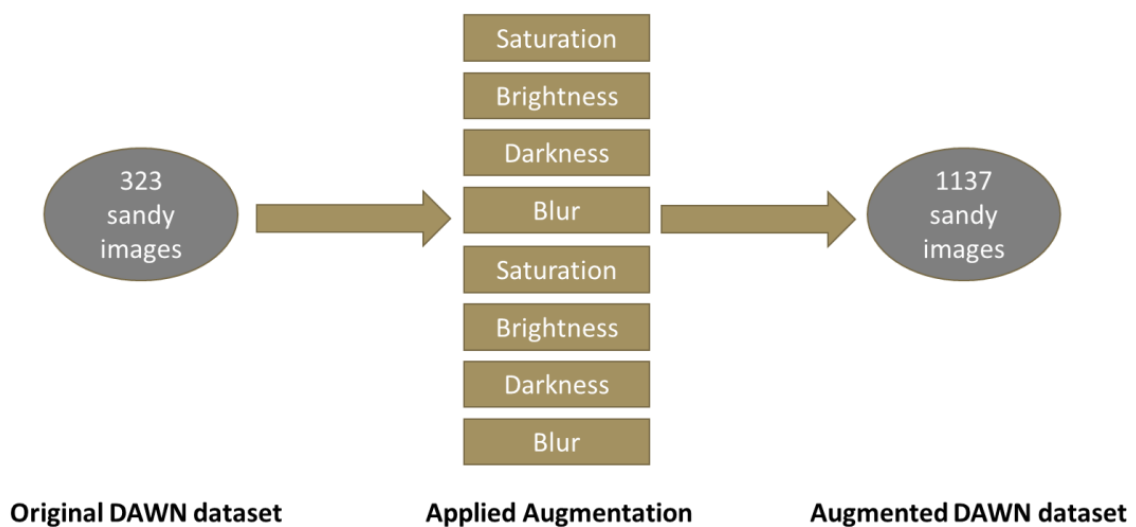


Figure 9. We increased the DAWN dataset from 323 images to 1137 sandy images.



Figure 10. Blur augmentation.



Figure 11. Saturation augmentation.



Figure 12. Brightness and darkness augmentation.

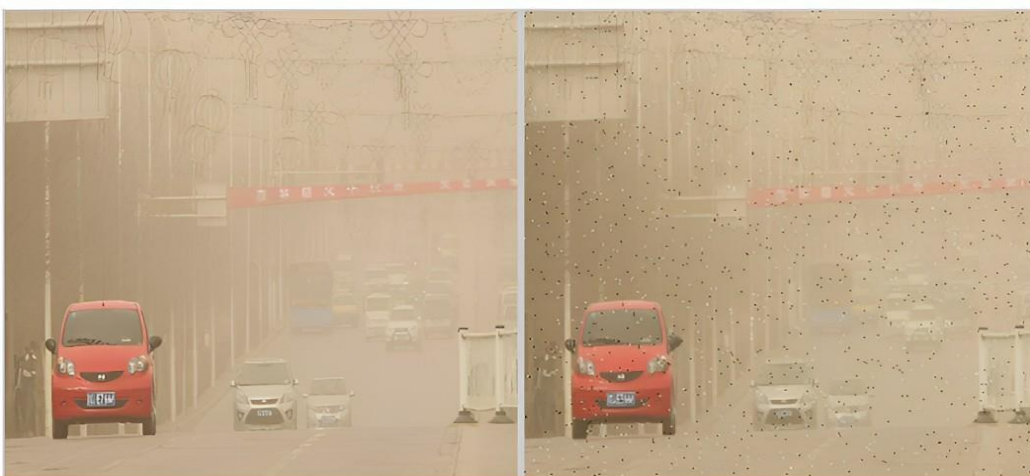


Figure 13. Noise augmentation.



Figure 14. Exposure augmentation.



Figure 15. Hue augmentation.



Figure 16. Grayscale augmentation.

6. Augmented DAWN with YOLOv5 and YOLOv7 Results

The experimental scenario for the augmented DAWN was executed within the Google Colab environment, harnessing the computational power of a its GPU. The machine learning framework PyTorch served as the cornerstone of our methodology, enabling efficient model training and evaluation. The experimentation process lasted for 64 epochs, with each epoch comprising 16 batches of images. The scenario detection task focused on three classes, car, person, and bicycle, reflecting common entities encountered in real-world scenarios. We used YOLOv7 as our base architecture due to its outstanding performance compared to other object detection models. YOLOv7 is considered as a state-of-the-art architecture. It outperformed all known object detectors in speed in the range of 5 FPS to 160 FPS and had the highest accuracy of 56.8% AP among all the known real-time object detectors, with 30 FPS or higher on GPU V100.

After running the experiment of YOLOv7, SiLU achieved the best overall best performance amongst all the activation functions in the augmented dataset. We achieved an mAP of approximately 94% using YOLOv7 and the SiLU function to detect the three classes. This is the highest score compared with the other adverse weather object detection publications that are mentioned in Table 1. The chart showing precision, recall, and mAP can be seen in Figure 17. The top left chart shows the precision result, with a score of 92%, while the top right chart shows the recall result reaching 85%. The bottom chart shows the resulting mAP which reached 94%. The F1 score is presented in Figure 18. The F1 score combines precision and recall into a single metric, providing us with a balanced assessment of our results. Our remarkably high F1 score is a strong indicator of the model's ability to balance between precise object detection in sandy weather (limited false positives) and capture most of the actual objects in the scene (limited false negatives). This is particularly important in real-time sandy weather tasks where both accuracy and completeness matter. The predicted labels of the experiments are shown in Figure 19.

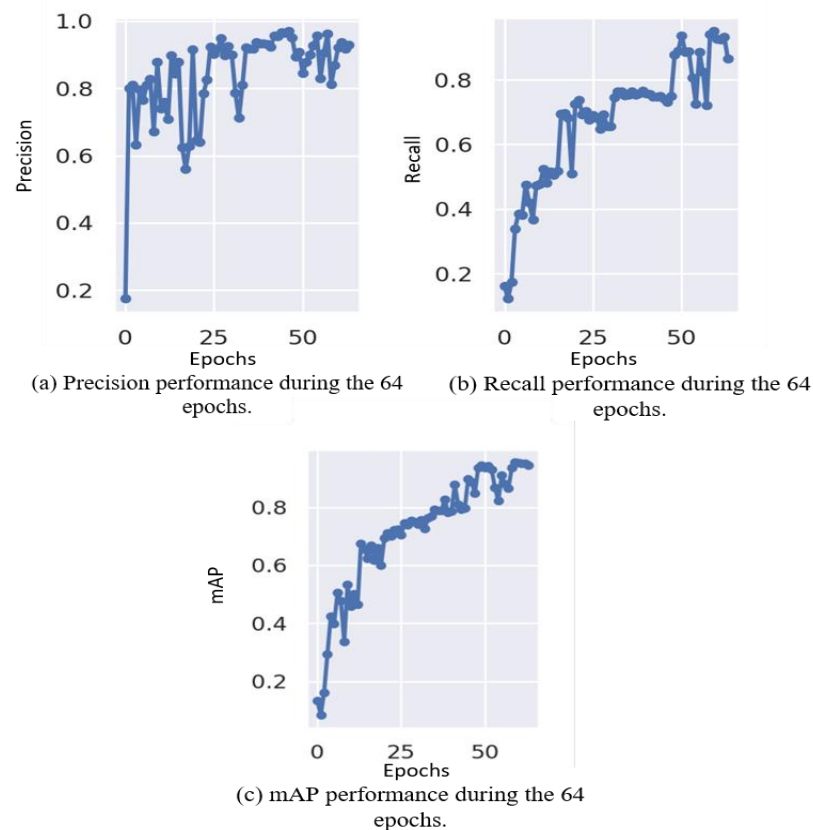


Figure 17. YOLOv7 results of our experiment.

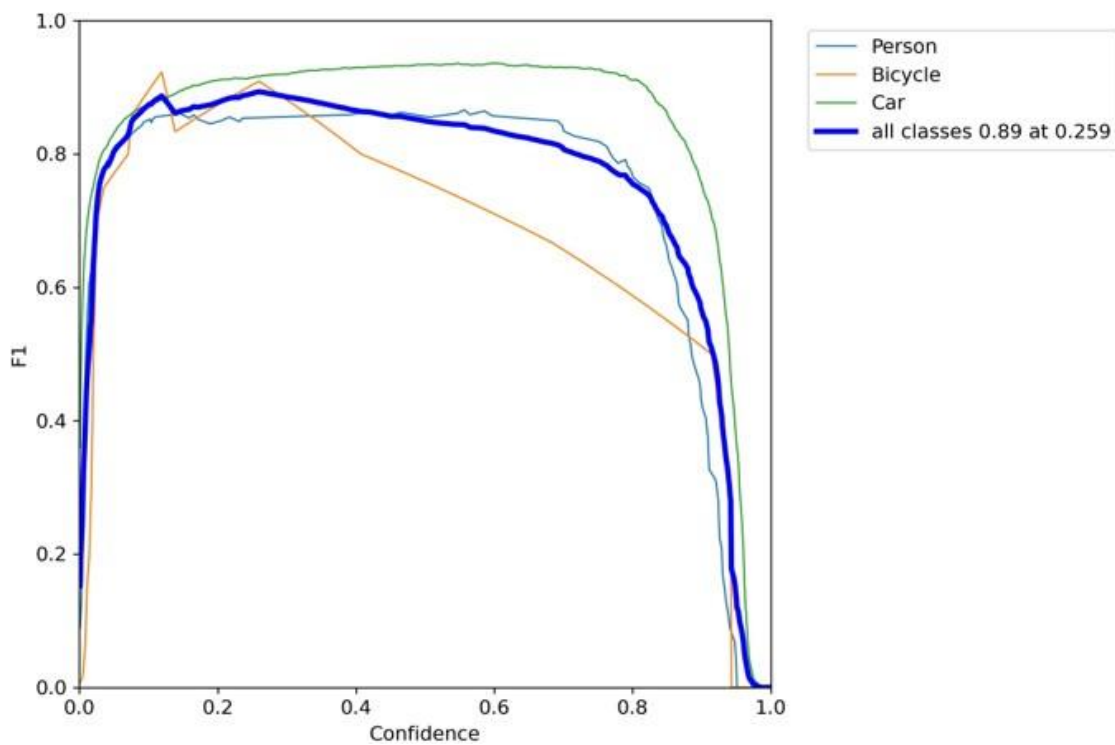


Figure 18. YOLOv7 F1 score.



Figure 19. YOLOv7 predicted objects using SiLU.

Choosing the right activation function is crucial for the model’s performance, as can be seen in Table 5. YOLOv7 with SiLU outperformed YOLOv7 with LeakyReLU with a significant performance. Using SiLU led to an increase in the mAP of the model by approximately 18 percentage points. This score is approximately 12 percentage points higher than that of the closest YOLOv5 model, which is YOLOv5l with SiLU. Regarding car class detection, all of the models, with their various activation functions, reached an mAP above 82%. The person class was quite similar to the car class, where the lowest achieved mAP was 83% using YOLOv5s with ReLU. Table 6 shows a summary of the

augmented DAWN results. From the activation functions point of view, SiLU achieved the highest mean average precision in all the scenarios using both the YOLOv5 and YOLOv7 models. As can be observed, relying on one activation function for all scenarios might lead to a drawback in the model's performance. If we compare the performance of the two YOLO versions with the original DAWN dataset and the augmented DAWN dataset, we can see that the highest mAP score was recorded for the augmented dataset using SiLU. However, in the original dataset, LeakyReLU had the highest mAP score. The results show that, in terms of mAP, there is no single model or single activation function that yields the highest performance in all scenarios. The selection of the model activation function is very subjective. It is based on the architecture design and the problem characteristics. In a paper published by Hnewa et al. [41], the authors stated, "There is a need for novel deep learning architectures and solutions that have adequate capacity for handling object detection under diverse conditions". This means that object detection still needs improvement in regard to experience with various environments. Our experiments support the aforementioned authors' statement.

Table 5. Results of augmented DAWN using YOLOv5 and YOLOv7.

Model	mAP	Function	Car mAP	Person mAP	Bicycle mAP
YOLOv5s	77%	SiLU	82%	84%	64%
YOLOv5s	73%	ReLU	83%	83%	54%
YOLOv5s	75%	LeakyReLU	85%	84%	55%
YOLOv5m	77%	SiLU	85%	87%	58%
YOLOv5m	77%	ReLU	87%	86%	58%
YOLOv5m	79%	LeakyReLU	86%	87%	65%
YOLOv5l	82%	SiLU	86%	87%	73%
YOLOv5l	78%	ReLU	88%	86%	60%
YOLOv5l	79%	LeakyReLU	86%	85%	66%
YOLOv7	76%	LeakyReLU	95%	85%	49%
YOLOv7	94%	SiLU	96%	89%	97%

Table 6. SiLU achieved the highest mean average precision in all scenarios using YOLOv5 and YOLOv7.

Activation Function	mAP
SiLU	82%
ReLU	76%
LeakyReLU	77%

7. Conclusions

Detecting objects with high accuracy is crucial for autonomous vehicles. Vehicles are required to be aware of every object in their surrounding environment and assure riders' safety. Enhancing object detection accuracy in sandy weather is particularly challenging due to several factors, such as poor object visibility and varying lighting conditions during storms. In this paper, we scrutinized three distinct activation functions (SiLU, ReLU, and LeakyReLU) with two YOLO models (v5 and v7). The highest achieved mAP score in our research was 94% with the SiLU activation function and the augmented DAWN dataset. Notably, our mAP was higher than the other documented outcomes in sandy weather contexts.

Furthermore, our experiments underscore the fact that no single activation function optimally suits all object detection scenarios. In some cases, YOLOv5 with LeakyReLU

produced the highest mAP, while in others, YOLOv7 with SiLU proved to be the top performer. For those using YOLOv5 in sandy weather, we recommend adopting the LeakyReLU function for greater accuracy. Conversely, if YOLOv7 is used, employing the SiLU function is highly recommended for optimal results.

8. Trends and Future Work

The area of object detection, specifically in the domain of AVs, is still under development and needs improvement. The eighth version of YOLO has been highlighted as the new state-of-the-art model for real-time object detection. A recent algorithm called You Only Learn One Representation (YOLOR) [42], which combines implicit and explicit knowledge, is a new approach that mimics the human capability for learning new things.

Fusion is a new trending architecture that can be used to enhance the outcome of the deep learning process. Model fusion works by unifying and integrating different inputs from different kinds of sources and produces a single outcome or a conclusion. It has been used in the medical field [43,44], texts and photos for hate speech classification [45], driver stress detection [46], and many other fields. Object detection, in an autonomous environment, is susceptible to the influences of different sources, such as images from cameras, signals from sensors, and text from road services. Creating a reliable model that is capable of combining data from all these sources can open the door for increased accuracy in detection and classification. Figure 20 shows different kinds of fusions: in Figure 20a, early fusion occurs, where different inputs are normalized and combined before inserting into the deep learning model for detection and classification; in Figure 20b, late fusion occurs, where each input data type has its own deep learning model for detection and classification, and then the results are combined to generate a single decision; and in Figure 20c, intermediate fusion occurs when the combination process unfolds on different layers. The authors of [47] used a fusion model to interpret information from two types of inputs: a camera and LiDAR. In [48], a sequential fusion called PointPainting was proposed.

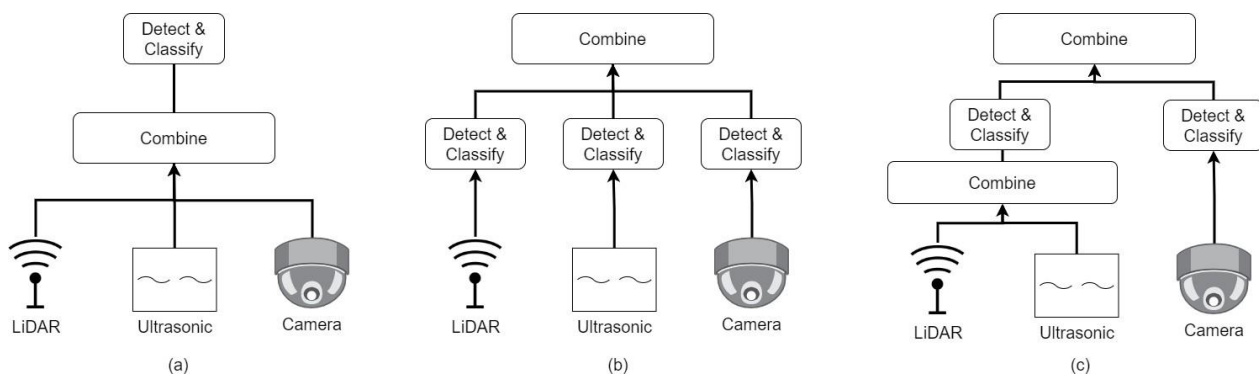


Figure 20. Types of fusion: (a) early fusion, (b), late fusion, (c) intermediate fusion.

Adverse weather is still a challenge in AV environments. Recent contributions focused on creating models that enhance the quality of the images in such weather [30,49–51]. This has opened the door for researchers in the AV field to develop promising models for the images fed into the CNN model.

In the future, we will extend our work to include other adverse weather conditions and the classes of detection to include new domains such as animals on the road. Additionally, since adding new classes requires a larger dataset, we will aim to have multiple sources (and formats) of data, enabling us to consider applying the fusions mentioned above. We believe there is a need for an object detection model that is suitable for all kinds of weather conditions and scenarios to ensure the highest accuracy for AV object detection. **Author**

Contributions: Conceptualization, N.A. and A.A.; methodology, N.A. and A.A.; software, N.A.; validation, A.A., V.T. and A.B.; formal analysis, N.A.; investigation, A.A., V.T. and A.B.; data curation, N.A.; writing—original draft preparation, N.A.; writing—review and editing, A.A., V.T. and A.B.; visualization, N.A.; supervision, A.A., V.T. and A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
2. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
3. Liu, T.; Sathaki, T. Faster R-CNN for Robust Pedestrian Detection Using Semantic Segmentation Network. *Front. Neurobot.* **2018**, *12*, 64. [[CrossRef](#)] [[PubMed](#)]
4. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; pp. 6569–6578.
5. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
6. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
7. Han, J.; Dai, H.; Gu, Z. Sandstorms and desertification in Mongolia, an example of future climate events: A review. *Environ. Chem. Lett.* **2021**, *19*, 4063–4073. [[CrossRef](#)] [[PubMed](#)]
8. Zijiang, Z.; Ruoyun, N. Climate characteristics of sandstorm in China in recent 47 years. *J. Appl. Meteor. Sci.* **2002**, *13*, 193–200.
9. Hadj-Bachir, M.; de Souza, P.; Nordqvist, P.; Roy, N. Modelling of LIDAR sensor disturbances by solid airborne particles. *arXiv* **2021**, arXiv:2105.04193.
10. Ferrate, G.S.; Nakamura, L.H.; Andrade, F.R.; Rocha Filho, G.P.; Robson, E.; Meneguette, R.I. Brazilian Road’s Animals (BRA): An Image Dataset of Most Commonly Run Over Animals. In Proceedings of the 2022 35th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Natal, Brazil, 24–27 October 2022; Volume 1, pp. 246–251.
11. Zhou, D. Real-Time Animal Detection System for Intelligent Vehicles. Ph.D. Thesis, Université d’Ottawa/University of Ottawa, Ottawa, ON, Canada, 2014.
12. Huijser, M.P.; McGowan, P.; Hardy, A.; Kociolek, A.; Clevenger, A.; Smith, D.; Ament, R. *Wildlife-Vehicle Collision Reduction Study: Report to Congress*; Federal Highway: Washington, DC, USA, 2017.
13. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010.
14. Dahl, G.E.; Sainath, T.N.; Hinton, G.E. Improving deep neural networks for LVCSR using rectified linear units and dropout. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 8609–8613.
15. Agarap, A.F. Deep learning using rectified linear units (relu). *arXiv* **2018**, arXiv:1803.08375.
16. Zeiler, M.D.; Ranzato, M.; Monga, R.; Mao, M.; Yang, K.; Le, Q.V.; Nguyen, P.; Senior, A.; Vanhoucke, V.; Dean, J.; et al. On rectified linear units for speech processing. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 3517–3521.
17. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16–21 June 2013; Volume 30, p. 3.
18. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the International Conference on Computer Vision, Las Condes, Chile, 11–18 December 2015; pp. 1026–1034.
19. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941.
20. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
21. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.

22. 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, 26 June–1 July 2016; pp. 779–788.
23. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
24. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
25. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
26. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
27. Humayun, M.; Ashfaq, F.; Jhanjhi, N.Z.; Alsadun, M.K. Traffic Management: Multi-Scale Vehicle Detection in Varying Weather Conditions Using YOLOv4 and Spatial Pyramid Pooling Network. *Electronics* **2022**, *11*, 2748. [[CrossRef](#)]
28. Wang, R.; Zhao, H.; Xu, Z.; Ding, Y.; Li, G.; Zhang, Y.; Li, H. Real-time vehicle target detection in inclement weather conditions based on YOLOv4. *Front. Neurobot.* **2023**, *17*, 1058723. [[CrossRef](#)]
29. Li, X.; Wu, J. Extracting High-Precision Vehicle Motion Data from Unmanned Aerial Vehicle Video Captured under Various Weather Conditions. *Remote Sens.* **2022**, *14*, 5513. [[CrossRef](#)]
30. Liu, W.; Ren, G.; Yu, R.; Guo, S.; Zhu, J.; Zhang, L. Image-adaptive YOLO for object detection in adverse weather conditions. In Proceedings of the AAAI Conference on Artificial Intelligence, Pomona, CA, USA, 24–28 October 2022; Volume 36, pp. 1792–1800.
31. Huang, S.-C.; Le, T.-H.; Jaw, D.-W. DSNNet: Joint Semantic Learning for Object Detection in Inclement Weather Conditions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 2623–2633. [[CrossRef](#)]
32. Sakaridis, C.; Dai, D.; Van Gool, L. Semantic Foggy Scene Understanding with Synthetic Data. *Int. J. Comput. Vis.* **2018**, *126*, 973–992. [[CrossRef](#)]
33. 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](#)]
34. Jung, H.-K.; Choi, G.-S. Improved YOLOv5: Efficient Object Detection Using Drone Images under Various Conditions. *Appl. Sci.* **2022**, *12*, 7255. [[CrossRef](#)]
35. Abdulghani, A.M.A.; Dalveren, G.G.M. Moving Object Detection in Video with Algorithms YOLO and Faster R-CNN in Different Conditions. *Avrupa Bilim Ve Teknol. Derg.* **2022**, *33*, 40–54. [[CrossRef](#)]
36. Zhang, C.; Eskandarian, A. A comparative analysis of object detection algorithms in naturalistic driving videos. In Proceedings of the ASME International Mechanical Engineering Congress and Exposition, Online, 1–5 November 2021; American Society of Mechanical Engineers: New York, NY, USA, 2021; Volume 85628, p. V07BT07A018.
37. Dazlee, N.M.A.A.; Khalil, S.A.; Abdul-Rahman, S.; Mutalib, S. Object detection for autonomous vehicles with sensor-based technology using yolo. *Int. J. Intell. Syst. Appl. Eng.* **2022**, *10*, 129–134. [[CrossRef](#)]
38. Kenk, M.A.; Hassaballah, M. DAWN: Vehicle detection in adverse weather nature dataset. *arXiv* **2020**, arXiv:2008.05402.
39. Zoph, B.; Cubuk, E.D.; Ghiasi, G.; Lin, T.Y.; Shlens, J.; Le, Q.V. Learning data augmentation strategies for object detection. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 566–583.
40. Volk, G.; Muller, S.; von Bernuth, A.; Hospach, D.; Bringmann, O. Towards Robust CNN-based Object Detection through Augmentation with Synthetic Rain Variations. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 285–292. [[CrossRef](#)]
41. Hnewa, M.; Radha, H. Object Detection Under Rainy Conditions for Autonomous Vehicles: A Review of State-of-the-Art and Emerging Techniques. *IEEE Signal Process. Mag.* **2020**, *38*, 53–67. [[CrossRef](#)]
42. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. You only learn one representation: Unified network for multiple tasks. *arXiv* **2021**, arXiv:2105.04206.
43. Lipkova, J.; Chen, R.J.; Chen, B.; Lu, M.Y.; Barbieri, M.; Shao, D.; Vaidya, A.J.; Chen, C.; Zhuang, L.; Williamson, D.F.; et al. Artificial intelligence for multimodal data integration in oncology. *Cancer Cell* **2022**, *40*, 1095–1110. [[CrossRef](#)] [[PubMed](#)]
44. Zhou, W.; Liu, W.; Lei, J.; Luo, T.; Yu, L. Deep Binocular Fixation Prediction Using a Hierarchical Multimodal Fusion Network. *IEEE Trans. Cogn. Dev. Syst.* **2021**, *15*, 476–486. [[CrossRef](#)]
45. Yang, F.; Peng, X.; Ghosh, G.; Shilon, R.; Ma, H.; Moore, E.; Predovic, G. Exploring deep multimodal fusion of text and photo for hate speech classification. In Proceedings of the Third Workshop on Abusive Language Online, Florence, Italy, 1 August 2019; pp. 11–18.
46. Mou, L.; Zhou, C.; Zhao, P.; Nakisa, B.; Rastgoo, M.N.; Jain, R.; Gao, W. Driver stress detection via multimodal fusion using attention-based CNN-LSTM. *Expert Syst. Appl.* **2021**, *173*, 114693. [[CrossRef](#)]
47. Zhao, X.; Sun, P.; Xu, Z.; Min, H.; Yu, H.K. Fusion of 3D LIDAR and Camera Data for Object Detection in Autonomous Vehicle Applications. *IEEE Sens. J.* **2020**, *20*, 4901–4913. [[CrossRef](#)]
48. Vora, S.; Lang, A.H.; Helou, B.; Beijbom, O. Pointpainting: Sequential fusion for 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4604–4612.
49. Wang, B.; Wei, B.; Kang, Z.; Hu, L.; Li, C. Fast color balance and multi-path fusion for sandstorm image enhancement. *Signal Image Video Process.* **2021**, *15*, 637–644. [[CrossRef](#)]

50. Shi, F.; Jia, Z.; Lai, H.; Song, S.; Wang, J. Sand Dust Images Enhancement Based on Red and Blue Channels. *Sensors* **2022**, *22*, 1918. [[CrossRef](#)]
51. 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.

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.