applied sciences

MDPI

Article

Improving YOLO Detection Performance of Autonomous Vehicles in Adverse Weather Conditions Using Metaheuristic Algorithms

İbrahim Özcan [1,*], Yusuf Altun [2] and Cevahir Parlak [3]

[1] Department of Computer Usage, Kütahya Dumlupınar University, 43020 Kütahya, Türkiye
[2] Department of Computer Engineering, Düzce University, 81620 Düzce, Türkiye; yusufaltun@duzce.edu.tr
[3] Department of Computer Engineering, Fenerbahçe University, 43020 Istanbul, Türkiye; cevahir.parlak@fbu.edu.tr
* Correspondence: ibrahimozcan84@gmail.com

Abstract: Despite the rapid advances in deep learning (DL) for object detection, existing techniques still face several challenges. In particular, object detection in adverse weather conditions (AWCs) requires complex and computationally costly models to achieve high accuracy rates. Furthermore, the generalization capabilities of these methods struggle to show consistent performance under different conditions. This work focuses on improving object detection using You Only Look Once (YOLO) versions 5, 7, and 9 in AWCs for autonomous vehicles. Although the default values of the hyperparameters are successful for images without AWCs, there is a need to find the optimum values of the hyperparameters in AWCs. Given the many numbers and wide range of hyperparameters, determining them through trial and error is particularly challenging. In this study, the Gray Wolf Optimizer (GWO), Artificial Rabbit Optimizer (ARO), and Chimpanzee Leader Selection Optimization (CLEO) are independently applied to optimize the hyperparameters of YOLOv5, YOLOv7, and YOLOv9. The results show that the preferred method significantly improves the algorithms' performances for object detection. The overall performance of the YOLO models on the object detection for AWC task increased by 6.146%, by 6.277% for YOLOv7 + CLEO, and by 6.764% for YOLOv9 + GWO.

Keywords: ARO; CLEO; DAWN dataset; deep learning; GWO; object detection; RTTS dataset; YOLOv5; YOLOv7; YOLOv9

check for updates

Citation: Özcan, İ.; Altun, Y.; Parlak, C. Improving YOLO Detection Performance of Autonomous Vehicles in Adverse Weather Conditions Using Metaheuristic Algorithms. *Appl. Sci.* **2024**, *14*, 5841. https://doi.org/10.3390/app14135841

Academic Editor: Yang Lu

Received: 28 May 2024
Revised: 20 June 2024
Accepted: 27 June 2024
Published: 4 July 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

Object detection studies can generally be divided into two main focus points: improvement of image processing methods and DL algorithms for one-stage or two-stage object detection. Image processing methods such as [1–6] generally involve performing operations on an image to make it better for object detection. DL algorithms for object detection, such as [7–16], are involved in improving the performance of DL architectures such as YOLO, Single Shot Detection (SSD), Faster RCNN, RetinaNet, CoupleNet CNN, SqueezeNet, VGG, ResNet, and DenseNet.

There are two approaches, one-stage or two-stage approaches, for object detection algorithms, as mentioned above. The one-stage approach means that object detection is performed by processing and analyzing all image pixels at once. The two-stage approach has two stages. In the first stage, region proposals are created, i.e., potential object regions are identified. In the second stage, object detection is performed in these regions. Both approaches have advantages over each other. While the single-pass approach emphasizes speed in object detection, the double-pass approach emphasizes higher accuracy. Since YOLO is faster and more efficient among the single-stage methods, such as SSD as in [17], and the detection speed is more important, especially for autonomous vehicles among the real-time applications, this paper aims to improve the performance of YOLO in adverse

weather conditions (AWCs). For these and similar reasons, YOLO is commonly used for object detection in the literature, such as [1–3,7,8,10,11,13,14,16]. For example, reference [1] proposes approaches that first perform image enhancements and then object detection. The TogetherNet module of [1] proposes a unified detection module for performing both image restoration and object detection using some versions of YOLO version x (YOLOx). In [2], the multi-scale retinex with color restoration (MSRCR) algorithm is used for pre-processing in vehicle detection in foggy weather, where YOLOv4 is used for object detection. The focus on foggy weather only in [2] does not provide sufficient data to measure the object detection performance of AWCs. In [3], a method for enhancing visibility and a robust approach for vehicle detection and tracking based on deep learning are Introduced. To solve the problem of low accuracy detection in AWCs, a visibility restoration method called Automatic White Balance Fused with Laplace Pyramid (AWBLP) is proposed. The DAWN dataset—containing objects such as cars, people, bicycles, buses, trucks, and motorcycles [18] under all AWCs, including fog, snow, rain, and sandstorms—is used, together with YOLOv3 and AWBLP. The YOLOv3 model used here was released In 2018. Considering the mAP (mean Average Precision) values in the study, the YOLOv3 model is slightly behind in performance. Another paper uses transfer learning for vehicle detection by adjusting the pre-trained architecture of YOLO-v5 through fine-tuning [10]. The DAWN dataset is used as the dataset. In [13], an algorithm was developed to better detect small objects. These algorithms were combined with YOLOv5 to create the Sf-YOLOv5 model. In [14], an improved YOLOv5 model for object detection in only foggy environments was proposed. A feature enhancement module (FEM) structure was created to detect foggy images, and this module enabled YOLOv5 to improve object detection performance.

Object detection studies in AWCs are one of the most emphasized topics in recent years. Researchers have focused on applying and improving object detection algorithms in autonomous vehicles with challenging problems [1,6,7]. In particular, the success of the vehicles in AWCs is important for performing safe and effective driving. Developments in computer vision and artificial intelligence have led to some successful results in difficult weather conditions for object detection, such as AWCs [1–4,6–9,12,14,19]. Among them, [4,7–9,12,14,15] mainly involve object detection using DL for AWCs. Reference [4] is based on the Realistic Single-Image Dehazing (RESIDE) dataset consisting of real and computer-generated hazy images. The aim of the study was to measure the limits of algorithms to remove haze in this RESIDE dataset and to compare them with each other using YOLOv3. But, it was used for the VOC_Foggy dataset containing only foggy weather conditions. Additionally, the used datasets do not contain trucks. In [7], Anchor-Free YOLOv4 was implemented to speed up vehicle detection. Using this model enabled a reduction in the required number of predictions per location from three to one. As a dataset, they used BDD-IW derived from the BDD100K dataset by adding fog to the images. Although it contains only sunny, rainy, and snowy weather conditions, there are no sandstorms conditions. Also, the old version YOLOv3 was used. The work in [8] proposes the Image-Adaptive YOLO (IA-YOLO) model to reduce the difficulty of object detection in AWCs. A differentiable image processing module is used to detect AWCs. But, it uses the VOC_Foggy dataset containing only foggy weather conditions. In [9], the authors compare the performance of DL algorithms for detecting road surface conditions of autonomous vehicles at night and in AWCs. For this purpose, the VGG, CNN, DenseNet, and SqueezeNet models are preferred for this comparison process. However, it does not contain sandstorm conditions, and it focuses on the road surface conditions at night in view of dry, wet, and snowy weather. There is no detection of objects such as buses, trucks, and persons. Moreover, it uses the older YOLOv2. In [12], it is emphasized that the performances of object detection algorithms decrease in AWCs. In order to improve the performance, it is emphasized that the image should be improved first. The proposed method combines the hazing module and the detection module and proposes an architecture called BAD-Net. But, it contains only foggy weather conditions. Additionally, the used datasets do not contain trucks. In [19], a technique of combining multiple deep learning models and data augmentation is proposed

to overcome the difficulty of object detection in AWCs. RetinaResnet50, YOLOv3, and SSD are used for the DAWN dataset. However, the performance results are lower according to this paper. As a result, the studies in the literature are either not comprehensive or appear to have low success in terms of AWCs.

Recently, metaheuristic algorithms have shown their effectiveness in applications in different fields, including path planning [20], image segmentation [21], feature selection problems [22], software engineering [23], artificial intelligence and machine learning [24], control engineering [25], structural engineering [26], and biological applications [27]. The Grey Wolf Optimizer (GWO) [28], Artificial Rabbit Optimizer (ARO) [29], and Chimpanzee Leader Selection Optimization (CLEO) [30], which are inspired by the hunting methods of wolves and the foraging methods of rabbits, are prominent metaheuristic algorithms for finding optimal solutions in complex search areas [25,31]. GWO is based on the hunting strategy and hierarchical structure led by alpha, beta, and delta grey wolves. ARO was developed by observing the foraging behavior of rabbits in nature. The CLEO algorithm is an optimization algorithm inspired by the social hierarchy and decision-making processes of chimpanzee communities, particularly the role of the alpha male chimpanzee as a leader. The aforementioned studies in the literature generally focus on the changing structure of DL algorithms to improve object detection performance based on DL. In addition to them, there are a few studies on the optimization of the structure, such as [15], where a boosted region proposal network (BRPN) is developed to overcome the shortcomings of Faster R-CNN. However, the datasets not containing AWCs are used to test BRPN. The loss function is improved by the metaheuristic algorithm GWO in the study. On the other hand, there are almost no studies that optimize the hyperparameters of the YOLO algorithms, other than [11], where a modified whale optimization algorithm (MWOA) is proposed by combining the whale optimization algorithm (WOA) and GWO to enhance the vehicle detection performance of YOLOv5. The algorithm optimizes the 12 default hyperparameters of the YOLOv5 model. However, it uses a dataset with normal weather conditions as the dataset; therefore, it does not address AWCs. As a result, the optimization of the YOLO hyperparameters are an open research topic for AWCs. Therefore, in the literature reviews, no studies on the optimization of hyperparameters of YOLOv5, YOLOv7, and YOLOv9 using metaheuristic optimization algorithms for object detection in autonomous vehicles under AWCs have been found.

Normally, the default hyperparameters are idealized for general usage. In addition, the parameter range is determined to be adjusted according to different applications. Although the default values of the hyperparameters can be successful for images without AWCs, there is a need to find the optimum values of the hyperparameters in AWCs. Since the number and range of hyperparameters are many and large, it is challenging to determine these parameters through trial and error. In particular, the optimization is crucial for object detection in autonomous vehicles under AWCs when compared to normal weather conditions. This is because, as can be seen from simulation studies, the parameter changes in AWCs greatly affect object detection performance. The effect of these metaheuristic algorithms on object detection performance on different datasets and different YOLO versions is investigated. This study proposes to improve the performance of object detection using YOLOv5, YOLOv7, and YOLOv9 algorithms in AWCs by using GWO, ARO, and CLEO metaheuristic optimization algorithms. In order to increase this performance, we aim to minimize the 1-mAP value as the objective function. The hyperparameters of the YOLO algorithms are employed to find optimum values for the datasets RTTS containing mainly normal weather conditions in addition to foggy conditions, as well as DAWN containing AWCs. YOLO object detection algorithms in AWCs have been significantly improved by GWO, ARO, and CLEO in this study.

2. Dataset Description and Applied Algorithms

2.1. Dataset Description

The DAWN and RTTS datasets with similar object labeling types are the preferred datasets in the literature. The DAWN dataset has data content that includes only road images and all adverse weather conditions. RTTS, on the other hand, has data content that includes only foggy and normal images. These two datasets are used to compare the performance of the applied methods.

DAWN: Vehicle Detection in Adverse Weather Nature Dataset: Few datasets address adverse weather conditions through a combination of synthetic weather and real-world images [18]. To overcome this shortcoming, Mourad A. Kenk and Mahmoud Hassaballah introduced a dataset called DAWN, which includes various AWCs to be used in vehicle detection in 2020 [18]. The DAWN dataset is a collection of high-resolution images and videos recorded under different weather conditions [18]. Roads in urban settings and rural settings, intersections, pedestrian crossings, and traffic signs are included in the dataset. Object detection, classification, and tracking can be performed using the dataset's labeled samples. Thanks to its wide scope and diversity, the DAWN dataset offers researchers and developers the opportunity to test and improve autonomous driving algorithms. This dataset is widely used for training and performance evaluation of deep learning models. In this study, we utilize the DAWN dataset for the detection of vehicles in AWCs. DAWN is a dataset specially created for AWCs. In this research, the version of the DAWN dataset customized for YOLO is used, which was obtained from the Roboflow website [32]. Figure 1 shows the labeling rates of the objects in the DAWN dataset. The labeled objects include people, bicycles, cars, motorcycles, buses, and trucks.

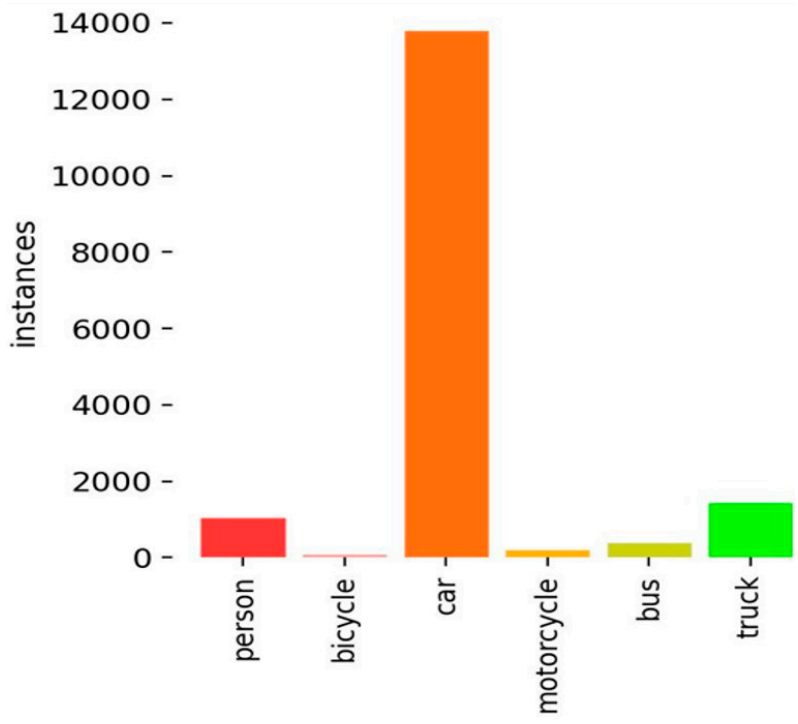


Figure 1. Sample numbers of labeled objects in the DAWN dataset.

This dataset is categorized as 2053 training, 197 validation, and 99 test data to be trained using the deep learning method. In this dataset, fog, snow, rain, and sandstorm images are randomly selected. 197 validation images are used to help determine the most efficient hyperparameters. A set of 99 test images that never entered training is used to evaluate the effectiveness of object detection. Figure 2 shows some images of the DAWN dataset.



Figure 2. DAWN dataset sample images.

Real-Time Transportation System (RTTS) Dataset: RTTS is the largest annotated detection dataset in hazy conditions [4]. It contains 4322 real-world hazy images, mostly in traffic scenes. There are five categories, including people, bicycles, motorcycles, buses, and cars. A total of 41,203 bounding boxes are labeled [12]. Public transportation systems are monitored and improved with the help of the RTTS dataset. This dataset typically includes location, speed, direction, and stop information for buses, trains, and other public vehicles. It contains various types of data such as GPS data, sensor data, and passenger counts. The RTTS dataset aims to use real-time transport data to make more accurate forecasts, optimize traffic flow, and improve passenger information systems. The RTTS dataset used for training in this study is the dataset produced for the YOLO algorithms on the Roboflow website [33]. The DAWN and RTTS datasets with similar labeling types are used. Figure 3 shows the labeling rates of the objects In the RTTS dataset. Figure 4 shows some images of the RTTS dataset. Additionally, the RTTS dataset containing normal weather conditions is also used for comparison with the DAWN dataset containing AWCs to determine the effects of proposed approach.

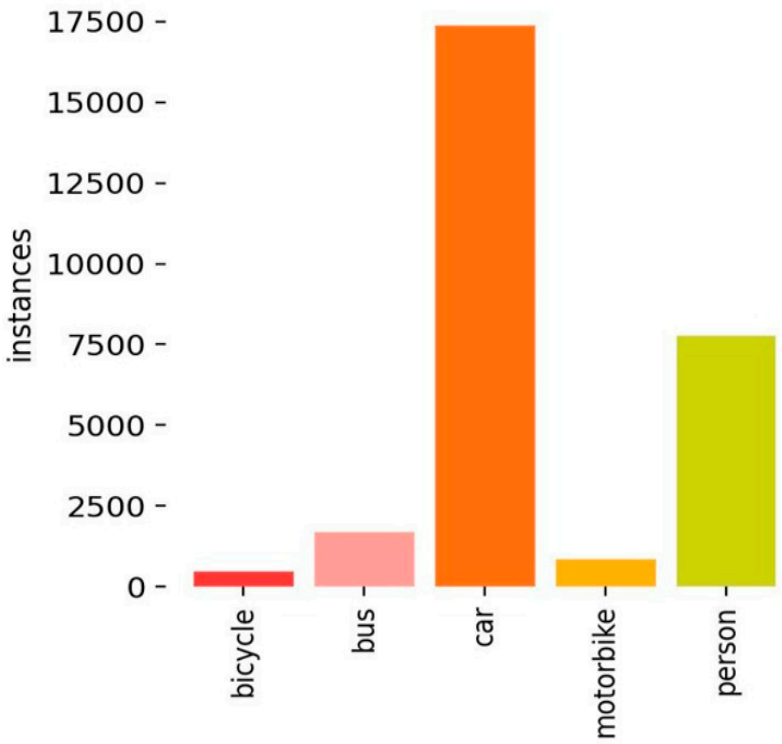


Figure 3. Sample numbers of labeled objects in the RTTS dataset.



Figure 4. RTTS dataset sample images.

2.2. Applied Algorithms

In this study, the DL algorithms YOLOv5, YOLOv7, and YOLOv9 were preferred for object detection. There are many studies on YOLOv5. YOLOv5 was chosen to compare the performance of the other two more recent algorithms in AWCs. Optimization algorithms were used to improve the hyperparameters of the YOLO models, especially the performance of the algorithms in AWCs. Various optimization algorithms can be used for this task. In this study, we chose GWO for object detection and the more recent successful ARO and CLEO metaheuristic algorithms for comparison.

YOLO version 5 (YOLOv5): YOLOv5 is part of the YOLO family for object detection, optimizing previous versions. YOLOv5 was developed by the Ultralytics company. Although there is no article in any peer-reviewed journal, it is preferred in scientific studies and shows effective results [5]. The steps related to the image to be detected using YOLOv5 are shown in Figure 5. The image arriving at the input layer is sent to the backbone for processing and feature extraction. The spine obtains feature maps with different dimensions and combines them with the neck to produce feature maps P3, P4, and P5. Finally, these maps are combined with the head to produce Bounding Boxes (Bboxes) that enclose the image in a rectangular frame [13].

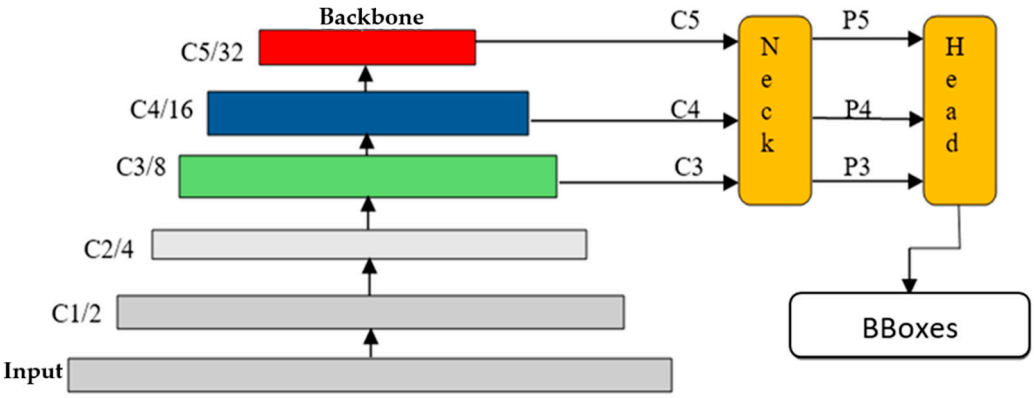


Figure 5. YOLOv5 default inference flow chart [13].

YOLO version 7 (YOLOv7): The YOLOv7 [34] algorithm has been widely recognized for its superior speed and accuracy in object detection, outperforming other established object detection models within the scope of 5 FPS to 160 FPS [35]. In the realm of urban infrastructure and transportation, YOLOv7 has been employed for road feature detection, pothole detection, and pedestrian–vehicle detection, demonstrating its significance in enhancing road safety and infrastructure maintenance [36,37]. YOLOv7 is one of the most recent official versions of YOLO released by the authors of the YOLO architecture in 2022. The architecture of YOLOv7 depends on a standard framework for object detection, which comprises three key elements: the backbone, neck, and head (see Figure 6). The backbone

functions as the feature extractor and is tasked with capturing hierarchical features from the input images. The backbone of the YOLOv7 architecture typically leverages a CNN (convolutional neural network) structure, such as CSPDarknet or EfficientNet, to efficiently extract features across various scales. The neck module, which follows the backbone, provides a collection of neural network layers that combine and mix features to the head section, which outputs the prediction. The head section takes the mixture of features from the neck and outputs a prediction.

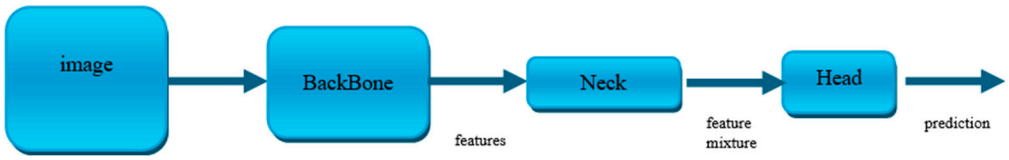


Figure 6. YOLOv7 model architecture [35].

YOLO version 9 (YOLOv9): YOLOv9 [38] is the latest YOLO algorithm, released in February 2024. When creating YOLOv9, two architectures were combined to improve its performance compared to previous models. Programmable Gradient Information (PGI) and a Generalized Efficient Layer Aggregation Network (GELAN) are combined to create YOLOv9. PGI, which prevents the loss of critical information in the trained data as it passes through multiple layers, makes learning more efficient. GELAN improves the performance of the model by performing optimization operations during the collection and processing of information in the layers. By combining these two architectures, YOLOv9 becomes a robust choice for object detection applications. The PGI and the related network architecture is given in Figure 7. The PGI consists of three main components: (1) a main branch that makes inferences, (2) an auxiliary reversible branch that provides reliable gradients for inference, and (3) multilevel auxiliary information. Multilevel semantic information can be learned to control main branch learning. As seen in Figure 7d, PGI uses only the main branch in its inference process, thus resulting in no additional cost. In deep learning methods, the other two components assist in solving or slowing down several important problems. As neural networks deepen, they encounter problems that require the auxiliary reversible branch to be developed. An information bottleneck will cause network deepening, resulting in unreliable gradients produced by the loss function. It is designed to handle the error accumulation problem caused by multiple prediction branches and the lightweight architecture caused by deep supervision.

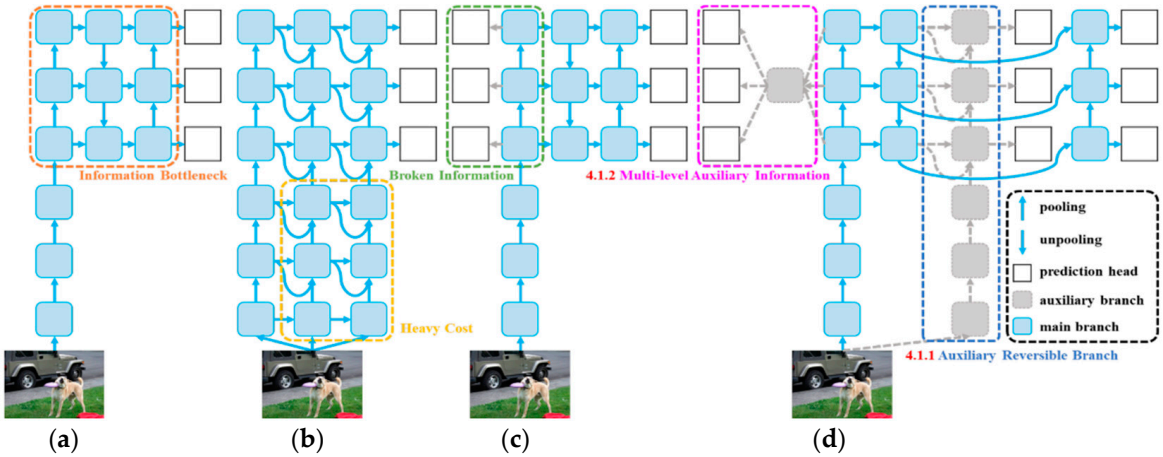


Figure 7. Methods and architectures for PGI networks. (**a**) Path Aggregation Network (PAN)) [39], (**b**) Reversible Columns (RevCol) [40], (**c**) conventional deep supervision, and (**d**) Programmable Gradient Information (PGI) [38].

GELAN's architecture is given in Figure 8. Enhancing Temporal Action Detection with Location Awareness (ELAN), which provides accurate and efficient detection of actions in videos, and Cross Stage Partial Network (CSPNet), which increases the efficiency and accuracy of convolutional neural networks, are combined to form GELAN. GELAN combines the strengths of these two architectures.

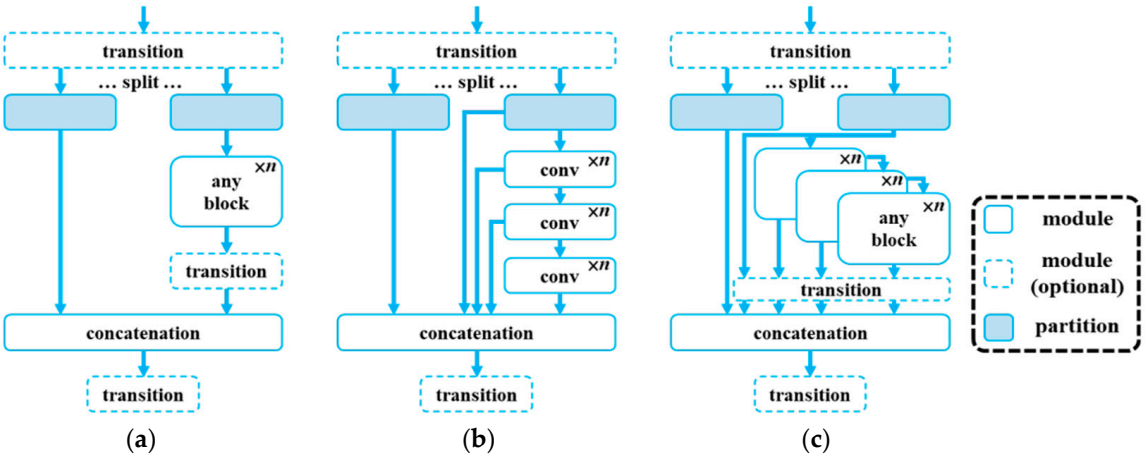


Figure 8. The architecture of GELAN: (**a**) CSPNet [41], (**b**) ELAN [42], and (**c**) GELAN [38].

Gray Wolf Optimizer (GWO): Based on grey wolf tactics and social behaviors, GWO is a metaheuristic optimization method. A balance between exploring the whole solution space and focusing on local regions is achieved by this algorithm by using alpha, beta, delta, and omega wolves to represent potential solutions within the search space [43]. Wolf positions are dynamically adapted in this process toward potential solutions, following the movement of predators tracking prey. The wolf positions are refined iteratively by the algorithm, leading to the optimal solution being reached with each iteration. Using these techniques, the GWO algorithm gradually narrows down the search space to find the most optimal solution. As a result, it is useful for optimizing a wide range of tasks. The iterative process of emulating the natural social hierarchy and hunting tactics of grey wolves within the GWO algorithm for tackling optimization problems is illustrated in the flowchart in Figure 9. During each iteration, candidate solutions, symbolized as wolf positions in the search space, undergo updates influenced by three primary roles: alpha, beta, and delta wolves, which represent the best, second-best, and third-best solutions identified thus far.

Artificial Rabbit Optimizer (ARO): The ARO was released as a new meta-heuristic optimization algorithm in 2022 [29]. Various optimization algorithms inspired by nature are developed to find the best solution. In ARO, the process of foraging and exploring the environment of rabbits is modeled and simulated in this way. Rabbits, by nature, avoid dangerous environments and at the same time explore new environments to find the best food source. Similarly, ARO provides an algorithm where optimal solutions can exist by discovering new search areas. In addition to machine learning, ARO has been applied for optimization in various fields such as finance, engineering, and logistics.

ARO offers a method that effectively performs optimization operations, inspired by rabbits searching for food in nature. Figure 10 shows the flowchart showing how the ARO algorithm works. The position and control parameters of the rabbits are loaded. The energy factor of the rabbits is calculated. As the number of iterations with a random oscillator increases, the extent of the energy factor A determines their search behavior. If A is greater than 1, a rabbit is randomly selected, and a foraging maneuver is performed. If A is less than 1, a tunnel is created, a random hiding place is chosen, and a random hiding maneuver is performed. After one of the two states is realized, the fitness is calculated, and the position is updated. The best solution was found until this state was updated. These steps are repeated until the stopping criterion is met. Finally, the best solution is returned.

ARO is an optimization method that can be used for various applications due to its iterative exploration and exploitation process. A solution space is navigated effectively, and optimal solutions are reached.

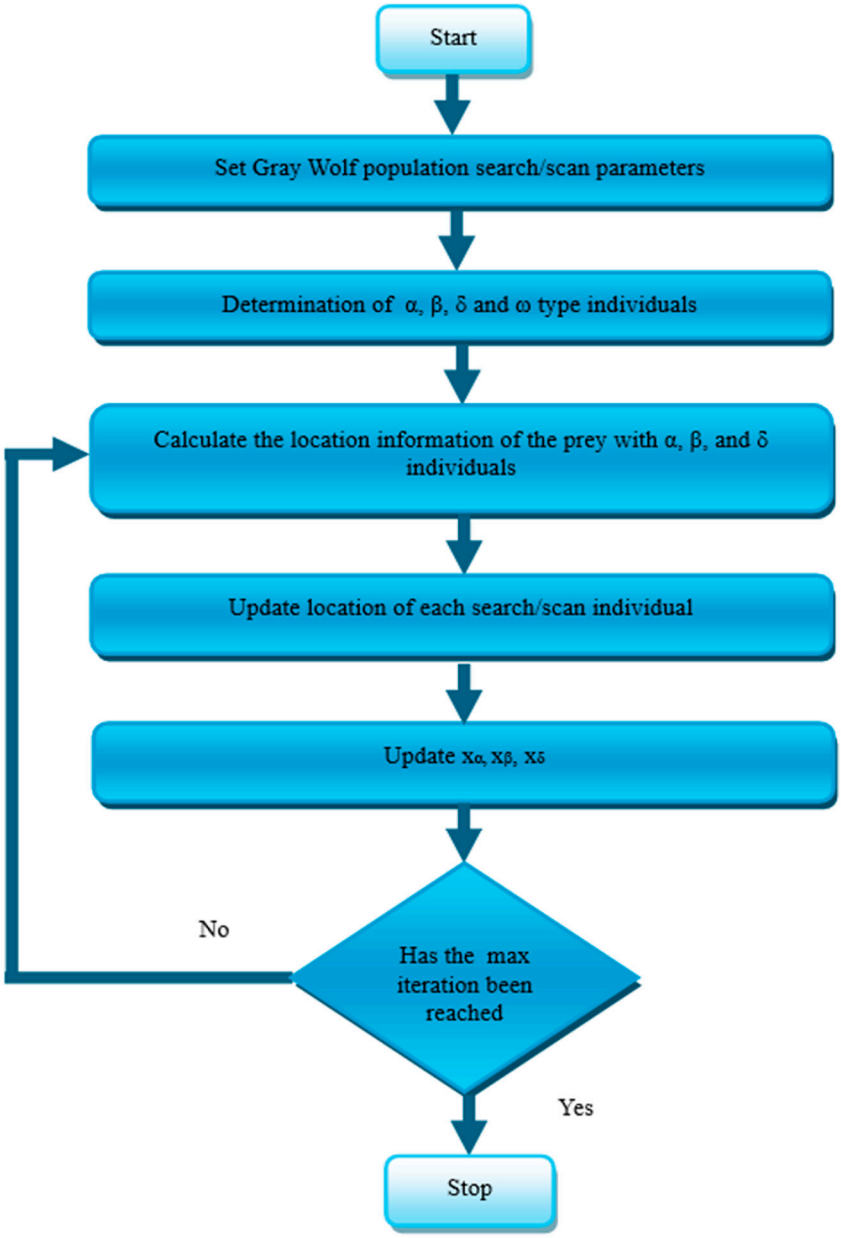


Figure 9. Flowchart of GWO algorithm [28].

Chimpanzee Leader Selection Optimization (CLEO): CLEO is one of the latest meta-heuristic optimization algorithms released in November 2022 [30]. It is inspired by the way the chimpanzee family chooses leaders [30]. The leader must be the alpha male. However, it is observed that the relationships of this chosen leader with other males and females play an important role in the CLEO algorithm [30]. The memorization ability of chimpanzees shows an inverted U-shaped performance improvement with age. Figure 11 shows the flow diagram of the CLEO algorithm showing these steps. The CLEO algorithm uses an iterative approach to find the optimal solution in the population. At each iteration, the algorithm calculates the fitness value for each male and female chimpanzee, selects the male and female chimpanzee with the highest fitness value, creates a new solution using the fitness values of the selected chimpanzees, calculates the fitness value of the new solution, replaces

the new solution with the least optimal solution in the current population, and stores the current best solution. The algorithm stops when the maximum number of iterations is reached or when the best solution meets a certain criterion.

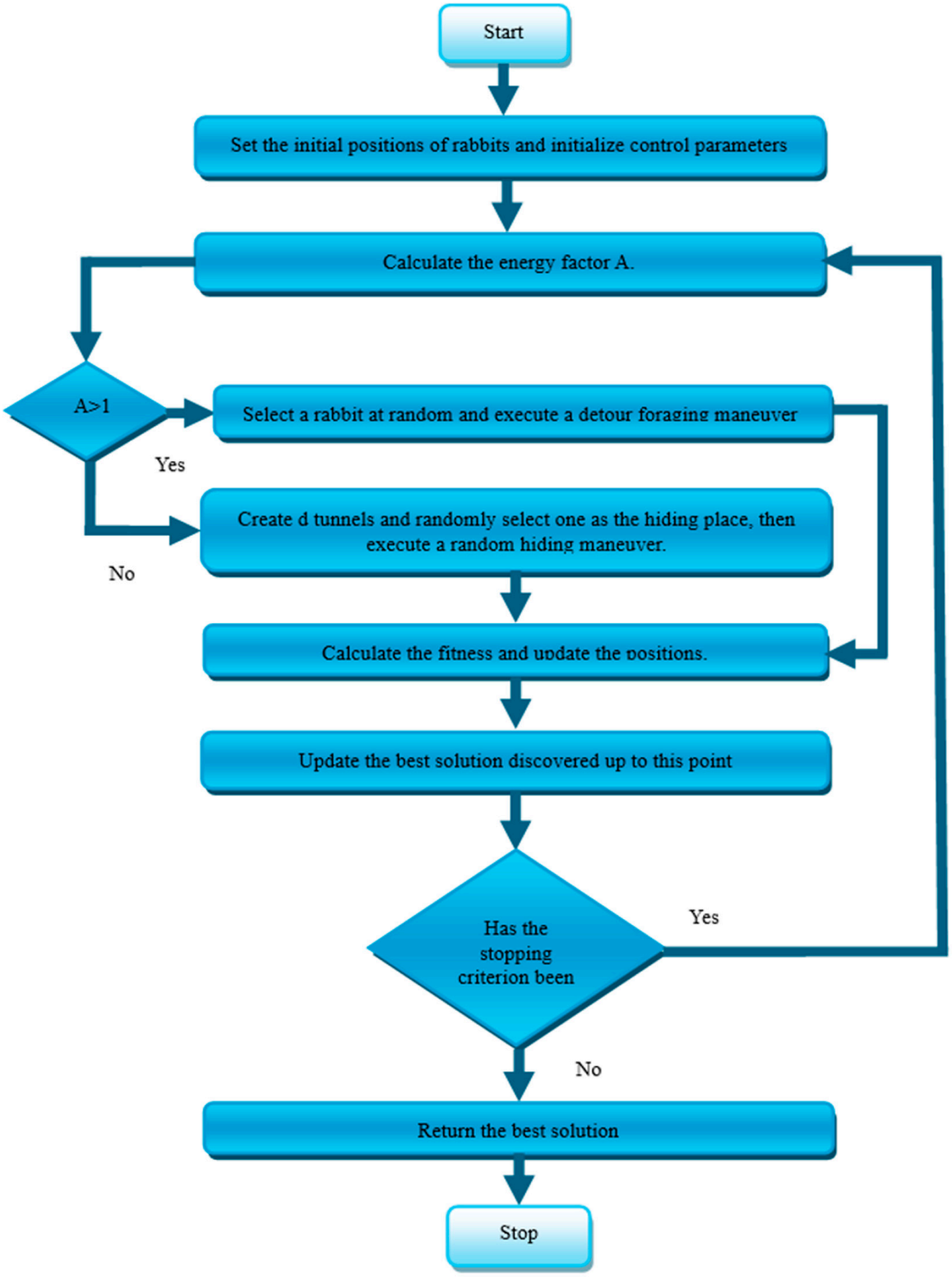


Figure 10. Flowchart of ARO algorithm [44].

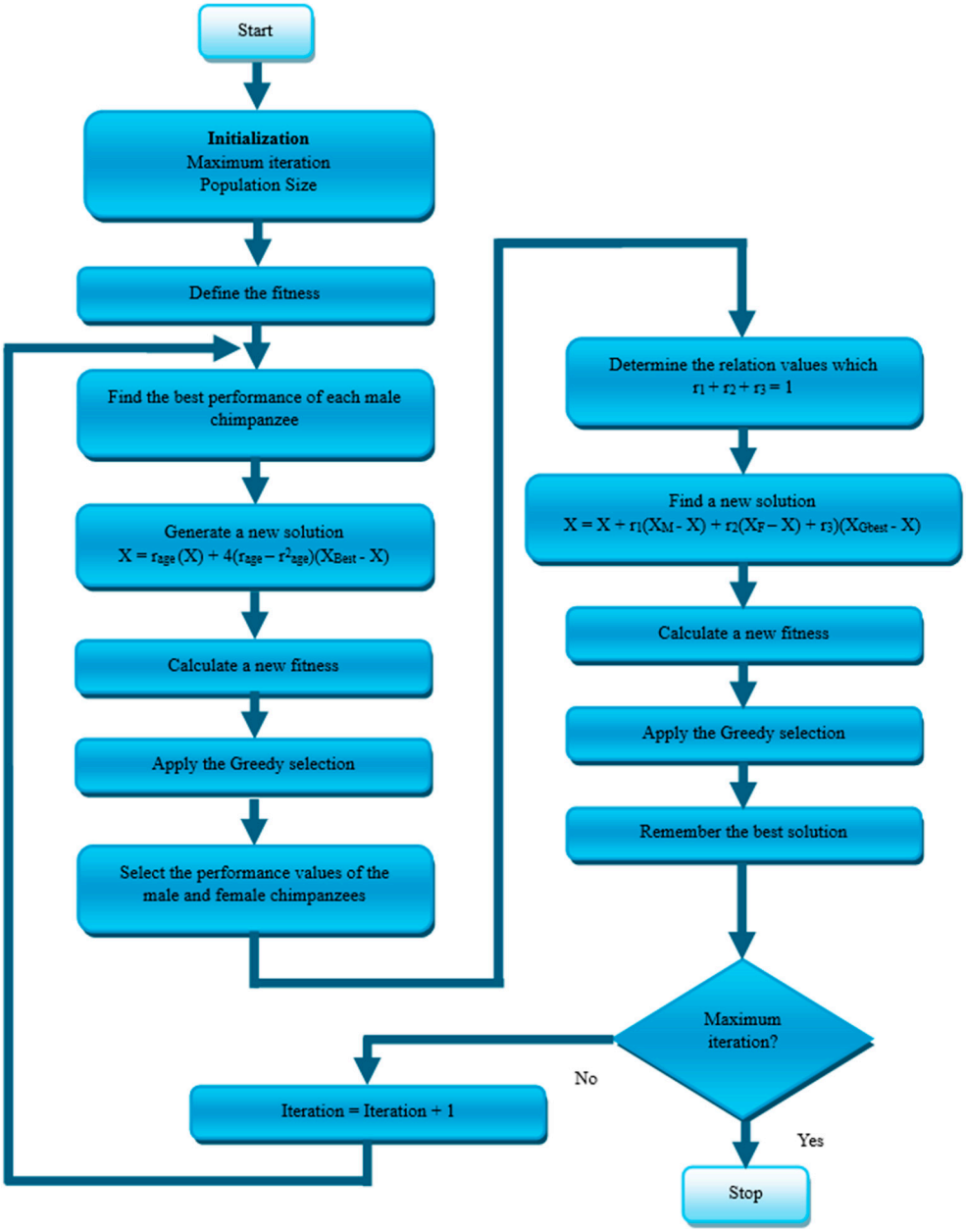


Figure 11. Flowchart of CLEO algorithm [30].

3. Proposed Methods

In this study, it is proposed to optimize the hyperparameters of the YOLOv5, YOLOv7, and YOLOv9 models mentioned in Section 2 to improve their object detection performance in AWCs. The GWO, ARO, and CLEO algorithms are proposed for these optimizations. In [11], the first 12 parameters of the hyperparameters of YOLOv5 are used for optimization. Generally, these first 12 parameters are preferred because they directly affect the training performance and accuracy of the deep learning models. These parameters include basic settings such as learning rate, momentum, weight reduction, number of boxes, number of classes, and loss weights. Since YOLOv5, YOLOv7, and YOLOv9 use the same infrastructure, the hyperparameters and value ranges are the same. Figure 12 shows the overall workflow process of this study. In our study, the first 12 hyperparameters of YOLOv5, YOLOv7, and YOLOv9 are used for optimization using GWO, ARO, and CLEO. Table 1 provides information about these hyperparameters.

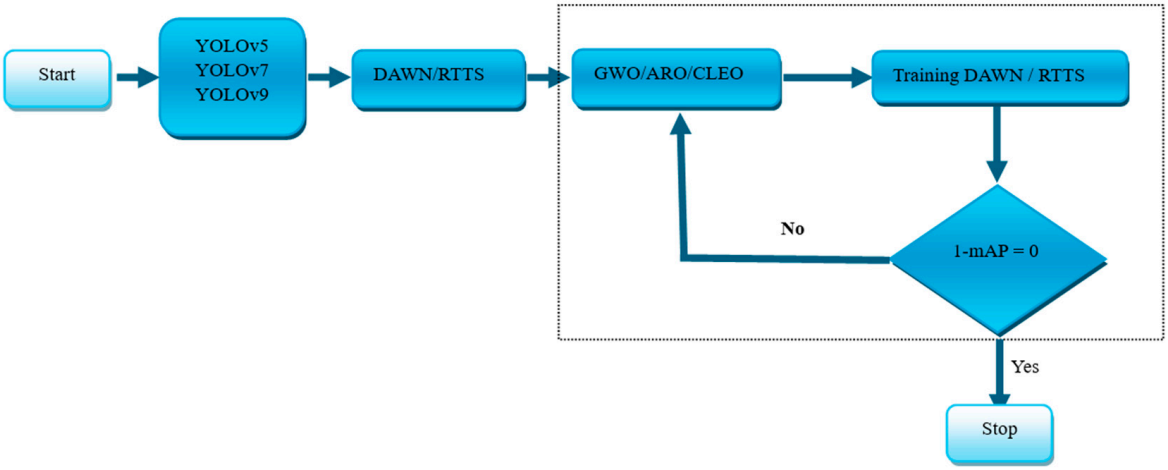


Figure 12. The workflow proposed in this study.

Table 1. The 12 hyperparameters of YOLOv5, YOLOv7, and YOLOv9.

Hyperparameter	Description	Lower Limit	Upper Limit
lr0	Initial learning rate	0.0010	0.1000
lrf	Final OneCycleLR learning rate	0.1000	0.9000
weight_decay	Optimizer weight decay	0.0005	0.0100
Momentum	SGD momentum	0.8000	0.9900
warmup_momentum	Warmup initial momentum	0.8000	0.9500
warmup_epochs	Warmup epochs	0.0000	1.0000
Warmup_bias_lr	Warmup initial bias lr	0.0000	0.1000
box	Box loss gain	0.0200	0.2000
obj	Object loss gain	0.2000	4.0000
cls_pw	Class BCELoss positive_weight	0.5000	2.0000
cls	Class loss gain	0.2000	4.0000
Obj_pw	Object BCELoss positive_weight	0.5000	2.0000

3.1. Training and Evaluation

Firstly, YOLOv5, YOLOv7, and YOLOv9 are trained using the default parameters on the DAWN and RTTS datasets. To compare the object detection performance, YOLOv5, YOLOv7, and YOLOv9 are trained using the hyperparameters optimized using GWO, ARO, and CLEO. The way the flow chart shown in Figure 12 works is as follows.

At the beginning, a 50-epoch training procedure is performed using the YOLOv5, YOLOv7, and YOLOv9 algorithms with the default hyperparameters.

Then, using GWO, training is performed with the hyperparameters, which are renewed every 50 epochs. In the same way, the object detection results are obtained using the same steps using ARO. And in the same way, the object detection results are obtained using the same steps using CLEO. The loop continues until the result 1-mAP = 0 is obtained. Or, since training is performed on data containing AWCs, when the mAP value reaches a certain value, the loop is terminated by external intervention when similar mAP results are encountered continuously. This training gives a result regarding the accuracy and efficiency of detecting vehicles in AWCs. These training results show that the object detection performance of YOLOv5, YOLOv7, or YOLOv9 using the DAWN and RTTS datasets, which contain AWCs but have some differences, varies depending on the dataset and the deep learning algorithms.

3.2. Performance Metrics

In order to evaluate the efficiency of the object detection model in YOLOv5, YOLOv7, and YOLOv9, the mean average precision (mAP) value is used in performance measurements in this study. The mAP value is one of the preferred metrics for performance

measurements in instantaneous object detection in autonomous driving. In addition, there are studies that also consider the F1 score value for object detection in datasets with imbalances in labeled data. The F1 score is determined as the harmonic mean of the precision and recall values. In the open-access datasets used in our study, since the data are unbalanced, another performance measure, accuracy, is not taken into account in the performance measurement, as it can be misleading.

While the algorithms detect objects, four different situations arise. True Positives (TPs) describe when the object is correctly predicted, True Negatives (TNs) describe when an existing object is incorrectly predicted, False Positives (FPs) describe when a non-existent object is predicted as if it exists, and finally, False Negatives (FNs) describe when the object exists but cannot be detected. These values allow the calculation of precision, recall, and the F1 score.

Precision: The ratio of the correct prediction of the samples taken to all the samples taken is called precision. Mathematically, precision is expressed by the equation provided in (1).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

Recall (Sensitivity): The ratio of TP predictions to all relevant samples is called recall. Mathematically, Recall is expressed by the equation provided in (2).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

F1 Score: The F1 score is the harmonic mean of the precision and recall and provides a balanced assessment of a model's performance while accounting for both false positives and false negatives. Mathematically, the F1 score is expressed by the equation provided in (3).

$$\text{F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

mAP (Mean Average Precision): The most important parameter used to measure and compare the performance of object detection algorithms, especially for vehicle and pedestrian detection, is mAP (mean average precision). The success of the model in detecting different objects is important. mAP determines the average precision performance of the model in all classes. This data show the general object detection performance of the model. Mathematically, mAP is expressed by the equation provided in (5).

$$\text{AP} = \frac{1}{\text{m}} \sum_{\substack{\text{r}=0.0 \\ \text{step } 0.1}}^{1.0} \text{p}(\text{r}) \tag{4}$$

$$\text{mAP} = \frac{1}{\text{k}} \sum_{\text{i}}^{\text{k}} \text{AP}_\text{i} \tag{5}$$

3.3. Implementation Environment

V100 GPU of Google Colab's powerful GPUs and high RAM were chosen to train the model. While images were included in the training, they were resized to 640 × 640 for the YOLOv7 standard. The batch size was set to 8. Training for 50 epochs using the default hyperparameters took 1 h and 7 min. Training with the hyperparameters optimized using GWO continued until the best results were found. Similarly, training with ARO continued until the best results were found. The code implementations of all the algorithms were developed in Python. The Python codes of the YOLOv5, YOLOv7, YOLOv9, GWO, ARO, and CLEO algorithms were taken from official repositories. This ensured the reliability and applicability of the results.

The computer system used to carry out these studies had a Core i7 10th Generation, Nvidia RTX 3060 6 GB GDDR6, 32 GB DDR4 3.2 GHz RAM, and 1 TB SSD.

4. Results and Discussion

4.1. Performance of YOLO Models on DAWN Dataset

Table 2 shows the training results of the YOLOv5, YOLOv7, and YOLOv9 models with original hyperparameters, GWO, ARO, and CLEO optimization algorithms in mAP type using the DAWN dataset. Figure 13 shows the graphical results of all training using the DAWN dataset after 50 epochs. The results of other studies using the DAWN dataset mentioned in this paper are also given. When the table is examined, it is seen that the training results of the YOLOv5, YOLOv7, and YOLOv9 models with the original hyperparameters are 60.20%, 68.50%, and 68.00%, respectively. Different results are seen when the training results after optimizing the hyperparameters of the YOLO models using the optimization algorithms are examined. For example, while CLEO provides the best performance improvement in YOLOv5 and YOLOv7, GWO is the best for YOLOv9. YOLOv5 + CLEO has a 63.90% mAP value with a 6.146% performance improvement. YOLOv7 + CLEO has 72.80% mAP with a 6.277% performance increase. YOLOv9 + GWO has 72.60% mAP with a 6.764% performance increase. According to Table 2, while the best mAP value is YOLOv7 + CLEO, the best performance improvement is realized with YOLOv9 + GWO in the YOLOv9 algorithm. Table 3 shows the precision, recall, and F1 score values obtained from the confusion matrix (e.g., "Figure 14"). Figure 14 is a confusion matrix value resulting from the training of only one algorithm. The values in Table 3 are generated by calculating the values of the 12 confusion matrix values one by one. These values are calculated separately for the six objects in the DAWN dataset, and different results are obtained for each applied model.

Table 2. The mAP (%) comparison of the DAWN dataset.

Model	DAWN (mAP (%))
YOLOv3 + AWBLP [3]	47.79
pre-trained YOLO-v5 [10]	34.50
RetinaResnet50 [19]	32.75
YOLOv3 + RetinaResnet50 [19]	25.68
RetinaResnet50 + SSD [19]	26.20
YOLOv3 + RetinaResnet50 + SSD [19]	25.03
YOLOv5	60.20
YOLOv5 + GWO (Ours)	62.30
YOLOv5 + ARO (Ours)	60.10
YOLOv5 + CLEO (Ours)	63.90
YOLOv7	68.50
YOLOv7 + GWO (Ours)	70.60
YOLOv7 + ARO (Ours)	71.20
YOLOv7 + CLEO (Ours)	72.80
YOLOv9	68.00
YOLOv9 + GWO (Ours)	72.60
YOLOv9 + ARO (Ours)	70.80
YOLOv9 + CLEO (Ours)	70.50

An example of graphical data showing all the changes that occurred throughout the training is shown in Figures 15 and 16. When the values are analyzed according to the average of all objects in Table 3, the best model for the F1 score is YOLOv7 + ARO. Since the detection and positioning of the object are more important in autonomous driving, mAP values are primarily considered. Therefore, it is observed that the YOLOv7 + CLEO and YOLOv9 + GWO models are superior to the other models in object detection and localization.

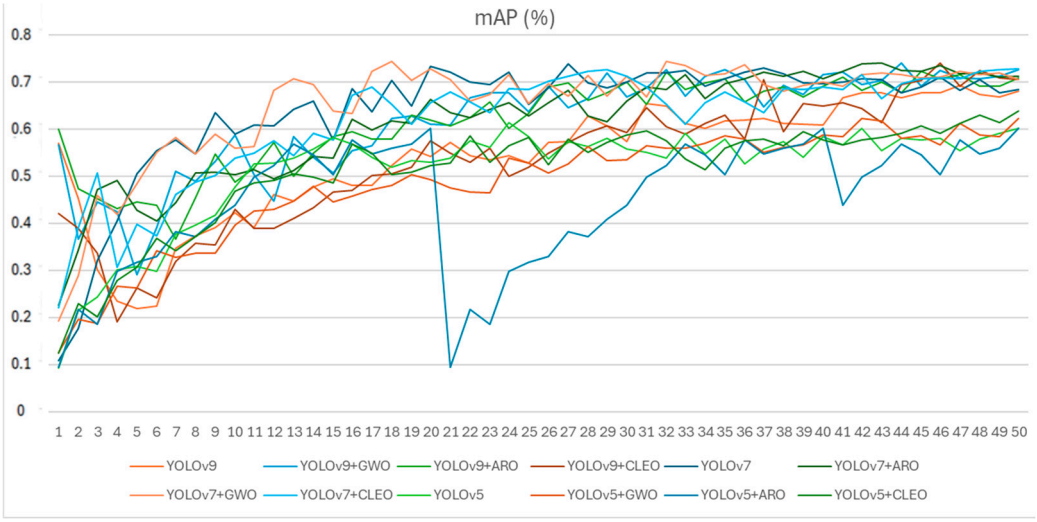


Figure 13. mAP results for 50 epochs on the DAWN dataset.

Table 3. Results using the DAWN dataset (%).

		Person	Bicycle	Car	Motorcycle	Bus	Truck
YOLOv5	Precision	88.70	0.00	49.69	95.74	94.73	91.80
	Recall	57.89	0.00	82.82	62.50	72.00	73.68
	F1 score	70.05	0.00	62.11	75.62	81.81	81.74
YOLOv5 + GWO	Precision	81.42	1.00	53.94	94.82	91.83	84.21
	Recall	61.29	50.00	82.82	67.07	76.27	77.10
	F1 score	69.93	66.66	65.33	78.56	83.32	80.49
YOLOv5 + ARO	Precision	86.76	0.00	51.87	96.96	88.88	88.23
	Recall	62.76	0.00	83.83	78.04	64.00	80.00
	F1 score	72.83	0.00	64.08	86.47	74.41	83.91
YOLOv5 + CLEO	Precision	85.33	89.47	51.57	94.11	95.34	92.85
	Recall	67.36	34.00	82.82	70.32	69.49	75.58
	F1 score	75.28	49.27	63.56	80.49	80.38	83.32
YOLOv7	Precision	90.66	1.00	51.86	97.33	96.49	87.01
	Recall	68.68	1.00	84.69	80.22	78.57	77.01
	F1 score	78.15	1.00	64.34	87.95	86.61	81.71
YOLOv7 + GWO	Precision	87.65	99.00	54.38	95.52	92.85	88.15
	Recall	71.71	1.00	87.87	70.33	91.23	81.70
	F1 score	78.88	99.49	67.18	81.01	92.03	84.80
YOLOv7 + ARO	Precision	88.09	99.00	55.00	95.34	92.15	87.84
	Recall	74.74	1.00	89.79	82.00	79.66	84.42
	F1 score	80.86	99.49	68.21	88.16	85.45	86.09
YOLOv7 + CLEO	Precision	88.75	99.00	53.12	95.89	96.34	87.83
	Recall	71.00	1.00	86.73	70.00	84.94	80.24
	F1 score	78.88	99.49	65.88	80.92	90.28	83.86
YOLOv9	Precision	82.43	1.00	51.57	96.96	96.15	91.17
	Recall	64.89	79.76	83.67	70.32	78.12	72.09
	F1 score	72.61	88.74	63.81	81.51	86.20	80.51
YOLOv9 + GWO	Precision	87.67	1.00	51.85	97.33	93.75	90.90
	Recall	67.36	79.76	84.84	73.00	76.27	80.45
	F1 score	76.18	88.74	64.36	83.42	84.11	85.35

Table 3. *Cont.*

		Person	Bicycle	Car	Motorcycle	Bus	Truck
YOLOv9 + ARO	Precision	91.17	1.00	49.41	98.64	96.15	95.89
	Recall	65.26	50.00	85.85	80.21	78.12	80.45
	F1 score	76.06	66.66	62.72	88.47	86.20	87.49
YOLOv9 + CLEO	Precision	83.95	1.00	54.24	98.46	84.37	85.33
	Recall	72.34	50.00	84.69	70.32	75.00	77.10
	F1 score	77.71	66.66	66.12	82.04	79.40	81.00

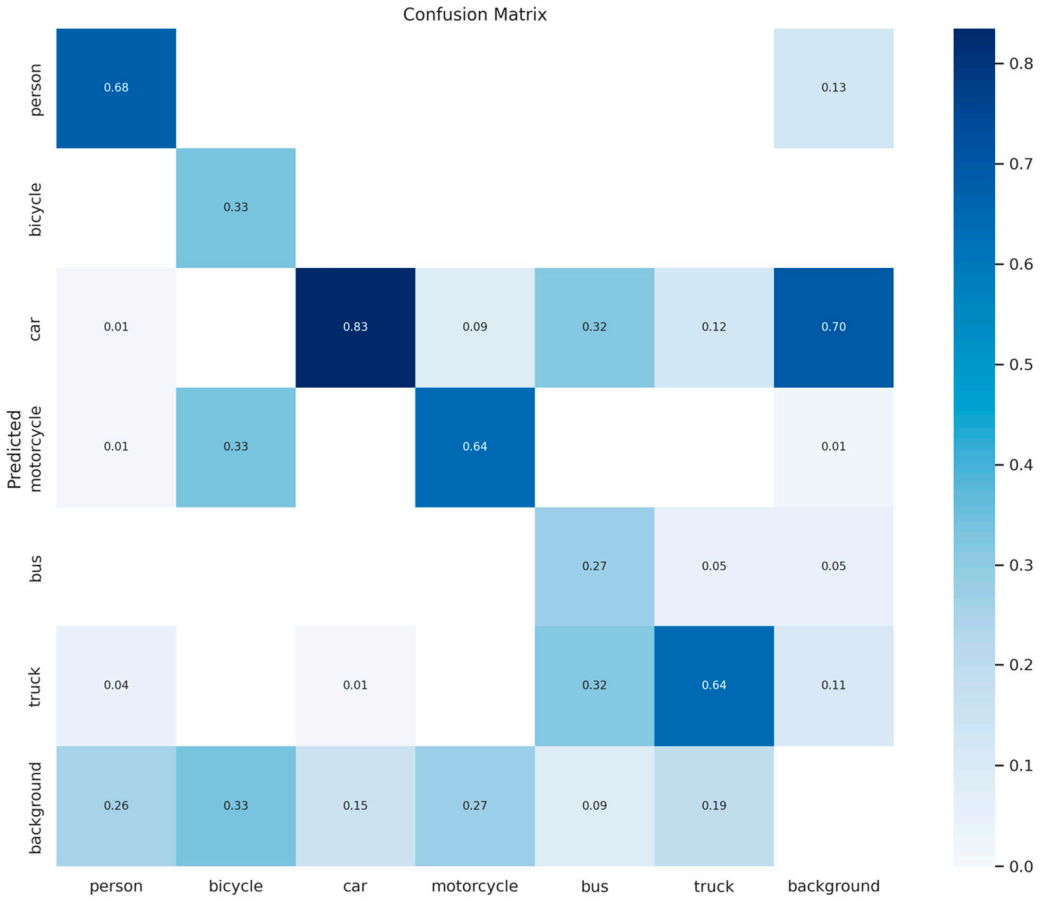


Figure 14. Confusion matrix diagram for YOLOv9 with CLEO parameters.

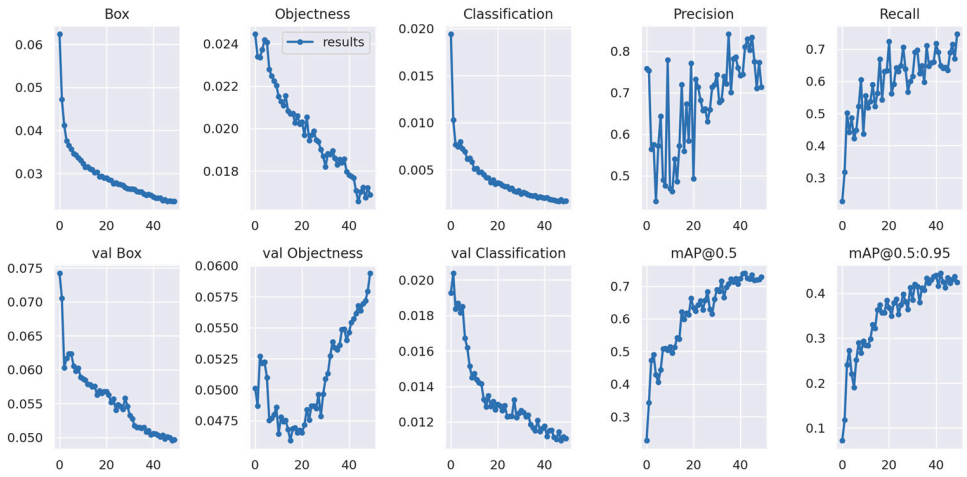


Figure 15. YOLOv7 + CLEO results.

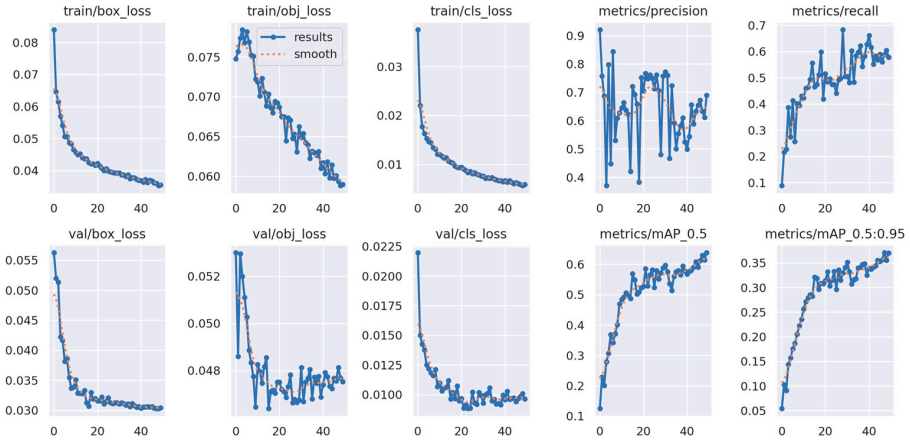


Figure 16. YOLOv5 + CLEO results.

4.2. Performance of YOLO Models on RTTS Dataset

Table 4 shows the training mAP results of the YOLOv5, YOLOv7, and YOLOv9 models using the RTTS dataset, with original and optimized hyperparameters with the GWO, ARO, and CLEO optimization algorithms. Table 4 also shows the results of other studies using the RTTS dataset mentioned in this paper. When the results are analyzed, it is seen that the mAP value of the YOLOv5-Fog model in [14] is higher than all the models of YOLOv5 and YOLOv7. The models with the best mAP values are the YOLOv9 models. The objects in the datasets and the number of labels of the objects are shown in Figure 3. Here, it is seen that the person tag is labeled considerably more than the other objects except for cars in Figure 3. Table 4 shows that when trained with the original hyperparameters of YOLOv5, YOLOv7, and YOLOv9 using the RTTS dataset, the mAP results are 75.60%, 76.80%, and 78.40%, respectively. It is observed that the optimization algorithms used with YOLOv5 do not affect the mAP result in YOLOv5 in any way. The YOLOv7 + GWO model realized a performance improvement of 0.13%, the YOLOv7 + ARO model realized a performance improvement of 0.65%, and the YOLOv7 + CLEO model realized a performance improvement of 1.04% compared to the original YOLOv7 model. The YOLOv9 + GWO model realizes a performance increase of 0.51%, the YOLOv9 + ARO model realizes a performance increase of 1.02%, and the YOLOv9 + CLEO model realizes a performance increase of 1.14% compared to the original YOLOv9 models.

Table 4. The mAP (%) comparison on the RTTS dataset.

Model	RTTS (mAP)
IA-YOLO [8]	52.00
TogetherNet [1]	61.55
Faster-RCNN [15]	51.30
RetinaNet [19]	53.10
YOLOv5-Fog [14]	77.80
YOLOv5	75.60
YOLOv5 + GWO (Ours)	75.60
YOLOv5 + ARO (Ours)	75.60
YOLOv5 + CLEO (Ours)	75.59
YOLOv7	76.80
YOLOv7 + GWO (Ours)	76.90
YOLOv7 + ARO (Ours)	77.30
YOLOv7 + CLEO (Ours)	77.60
YOLOv9	78.40
YOLOv9 + GWO (Ours)	78.80
YOLOv9 + ARO (Ours)	79.20
YOLOv9 + CLEO (Ours)	79.30

Compared to the original YOLOv7 model, the YOLOv9 + GWO model realizes a performance increase of 0.51%, the YOLOv9 + ARO model realizes a performance increase of 1.02%, and the YOLOv9 + CLEO model realizes a performance increase of 1.14% compared to the original YOLOv9 model. Also, Figure 17 shows the graphical results of all training using the RTTS dataset after 50 epochs. An example of graphical data showing all the changes that occurred throughout the training is shown in Figures 18 and 19.

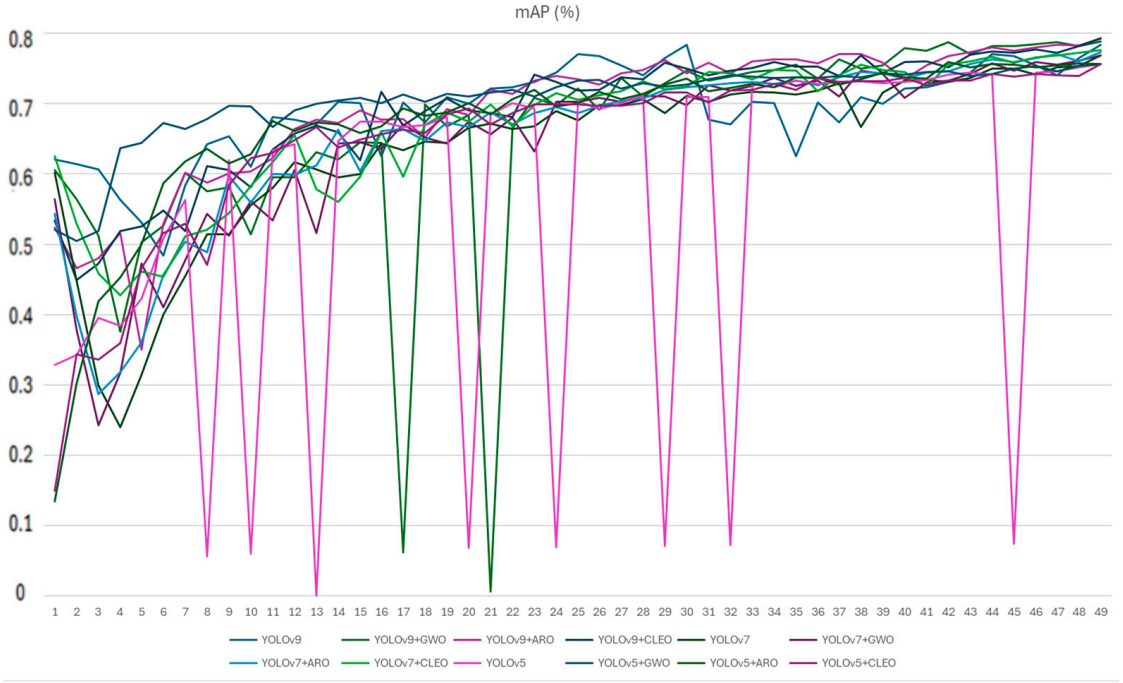


Figure 17. mAP results for 50 epochs on the RTTS dataset.

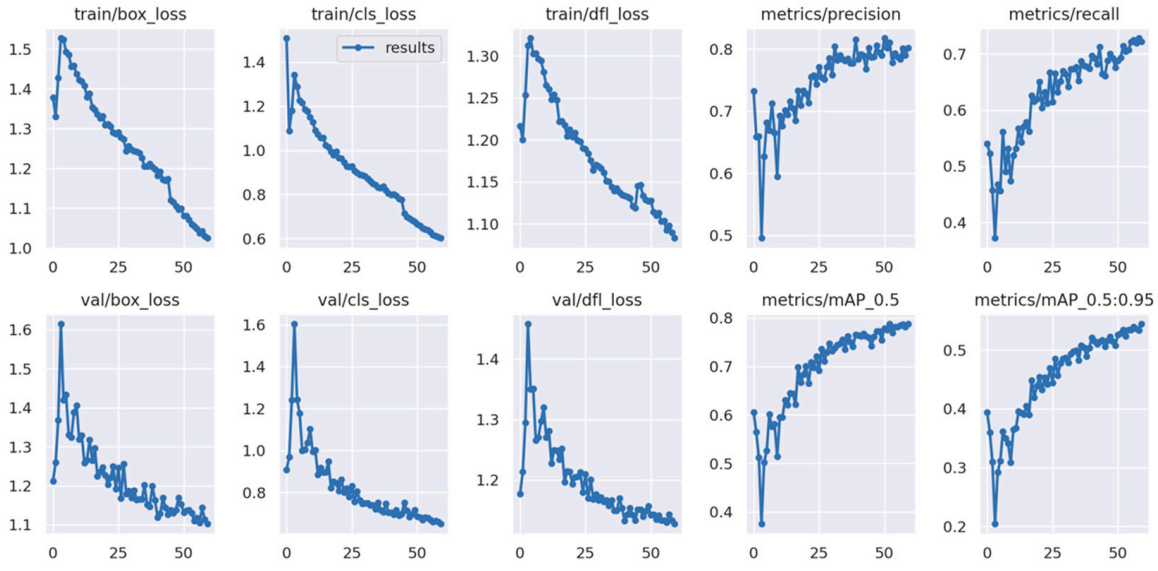


Figure 18. YOLOv9 + GWO results.

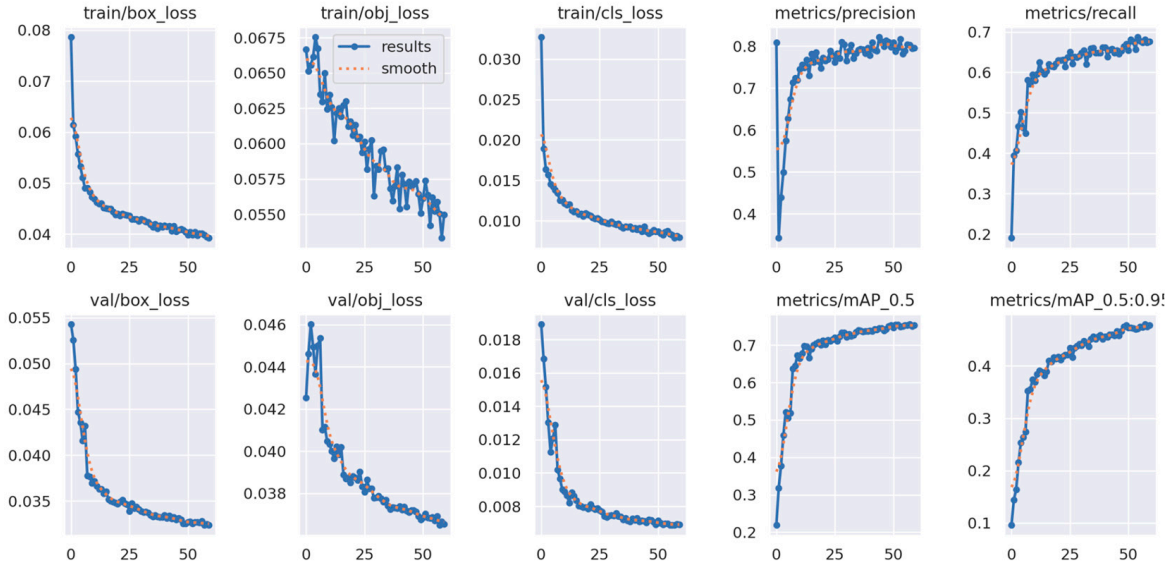


Figure 19. YOLOv5 + GWO results.

In our study, the increase in object detection performance with hyperparameter optimization shows that the intended goal was achieved, especially when including AWCs and only traffic roads. The differences are that we only trained on AWCs and used GWO, ARO, and CLEO for hyperparameter optimization. In [11], the 12 hyperparameters of YOLOv5 are used for object detection using the MWOA optimization algorithm. Since the used KITTI [45] dataset usually contains normal, clean images, it is not known how it will perform in AWCs. Therefore, it uses the dataset with normal weather conditions, and so it does not address AWCs. Additionally, it uses only the older version of YOLOv5.

After training the DAWN dataset with the default parameters, Figure 20 shows some images presented to the system for testing. In the first row and third column, some vehicles are not detected. In the first column of the second row, the oncoming vehicle is not detected. In other examples, the first row detects objects that are not in the fourth column. Some of the images entered into the system for testing purposes from the DAWN dataset trained with GWO-optimized hyperparameters are shown in Figure 21. Compared to Figure 20, the truck in the first row and fourth column is predicted and detected at a higher rate, and a vehicle that is not in the first row and fourth column is not labeled as a vehicle. In the second row and first column, the oncoming vehicle is labeled. The next set of test data images from the DAWN dataset trained with ARO-optimized hyperparameters are shown in Figure 22. Compared to Figure 20, all cars are detected and labeled in the sandstorm image in the first row and third column. The oncoming vehicle in the second row and first column and the truck in the fourth row and fourth column are not labeled as in Figure 20. The next test data images from the DAWN dataset trained with CLEO-optimized hyperparameters are shown in Figure 23. Compared to Figure 20, all cars are detected and labeled in the sandstorm image in the first row and third column. In the same image, an absent vehicle is labeled. Looking at the mAP values in Table 2, the smallest mAP value is 68.00% and the largest value is 72.60%, especially for YOLOv5 and YOLOv9. Looking at the labeled objects in the figures, it is observed that object detection operations are performed in accordance with the mAP values.



Figure 20. Test result with original hyperparameters.

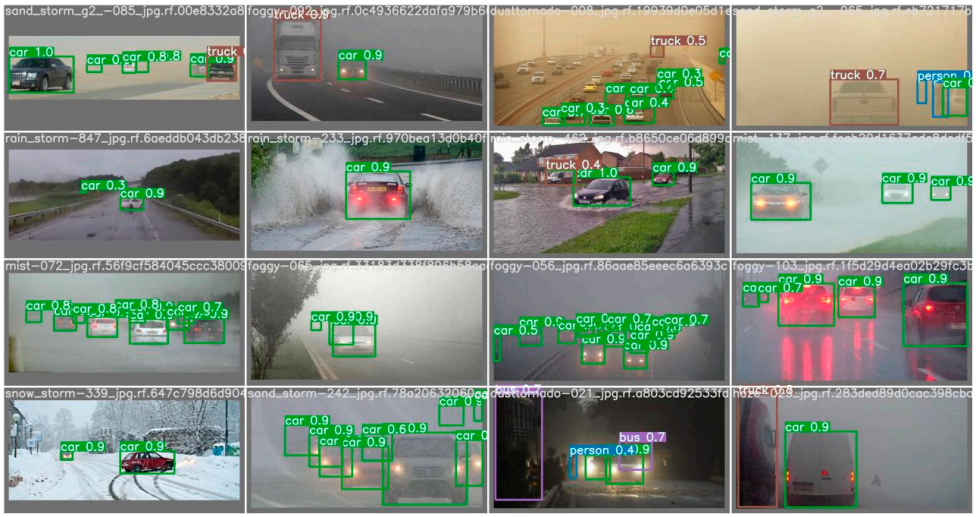


Figure 21. Test results with GWO optimizer hyperparameters.

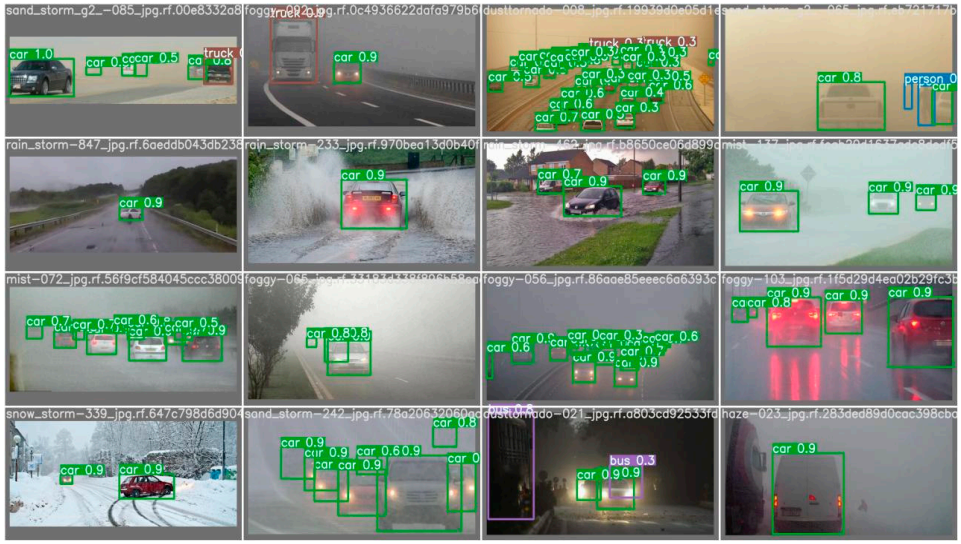


Figure 22. Test results with ARO optimizer hyperparameters.

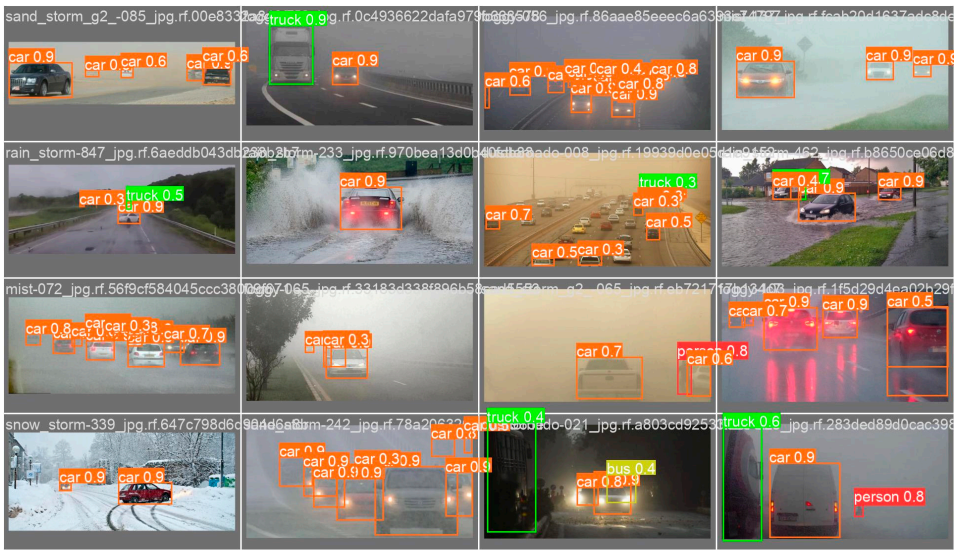


Figure 23. Test results with CLEO optimizer hyperparameters.

Table 5 shows the iteration rate, training time, advantages, and disadvantages of the applied methods for all training using the DAWN dataset, especially our study since it includes all AWCs.

Table 5. Object detection performance and efficiency comparisons.

Method	F1 Score	Iteration Rate (it/s)	50 Epoch Hours	Advantages	Disadvantages
YOLOv5	49,690	1.97	0.661	Provides faster responses for real-time applications.	Low accuracy in object detection
YOLOv5 + GWO	59,295	2	0.58	Ideal for applications requiring high accuracies	More processing power and energy consumption
YOLOv5 + ARO	52,025	1.06	0.63	Low training time	Low accuracy in object detection
YOLOv5 + CLEO	57,607	1.88	0.61	Provides faster response for real-time applications	Object detection accuracy is insufficient
YOLOv7	56,097	1.71	0.63	A balanced model, efficient in both speed and training time	Lower F1 score compared to other YOLOv7 variants
YOLOv7 + GWO	73,612	1.55	0.63	Provides a high accuracy with a good F1 score.	Relatively low speed—this may cause some delays in real-time applications
YOLOv7 + ARO	74,612	1.38	0.64	Ideal for applications requiring high accuracy	Low accuracy in object detection
YOLOv7 + CLEO	72,622	1.22	0.65	Provides a good balance for both speed and training.	The intermediate F1 score does not provide the highest accuracy
YOLOv9	65,927	1.06	0.65	Low training time	F1 score average is lower than other YOLOv9 variants.
YOLOv9 + GWO	67,890	1.11	0.66	Good F1 score and moderate speed	Training time may be longer than other YOLOv9 variants

Table 5. *Cont.*

Method	F1 Score	Iteration Rate (it/s)	50 Epoch Hours	Advantages	Disadvantages
YOLOv9 + ARO	64,988	1.42	0.67	High speed and high accuracy	Training period is slightly longer
YOLOv9 + CLEO	61,073	1.05	0.66	Offers a low training time and good balance	Low speed may cause performance degradation in some applications

Looking at the results in general, models optimized using YOLOv7 + CLEO or YOLOv9 + GWO, which have the highest mAP value, can be used in applications requiring high accuracies. The models used may vary depending on weather conditions. However, by using the YOLOv7 or YOLOv9 models, the best mAP results can be obtained regardless of the weather conditions. When the prediction images are examined in detail, it is clear that the YOLO models optimized separately using GWO, ARO, and CLEO improve the detection of vehicles, especially when using AWC datasets in dusty weather conditions. All three training configurations (GWO, ARO, CLEO, and original) showed success in object detection, regardless of weather conditions, including scenarios beyond dusty weather. These findings highlight the effectiveness of the YOLOv7 and YOLOv9 algorithms in detecting objects in AWCs and perform even better when combined with the GWO, ARO, or CLEO optimization algorithms. The results and original values of the first 12 hyperparameters of the YOLO versions 5, 7, and 9 models optimized using meta-heuristic optimization algorithms are shown in Tables 6 and 7.

Table 6. Optimized and original top 6 hyperparameters in YOLOV5, YOLOv7, and YOLOv9.

	lr0	lrf	Momentum	Weight_Decay	Warmup_Epochs
YOLOv5	0.0100	0.0100	0.9370	0.0005	3.0000
YOLOv5 + GWO	0.0124	0.0129	0.9781	0.0008	3.2113
YOLOv5 + ARO	0.0100	0.0100	0.937	0.0005	3.0000
YOLOv5 + CLEO	0.0115	0.0109	0.9556	0.0007	3.2030
YOLOv7	0.0100	0.1000	0.937	0.0005	3.0000
YOLOv7 + GWO	0.0123	0.1286	0.9781	0.0008	3.2113
YOLOv7 + ARO	0.0120	0.1176	0.9406	0.0006	3.0755
YOLOv7 + CLEO	0.0114	0.0125	0.9638	0.0006	3.0110
YOLOv9	0.0100	0.0100	0.9370	0.0005	3.0000
YOLOv9 + GWO	0.0106	0.0107	0.9472	0.0006	3.0526
YOLOv9 + ARO	0.0118	0.0129	0.0006	3.1323	0.8188
YOLOv9 + CLEO	0.0116	0.0129	0.9737	0.0006	3.0202

Table 7. Optimized and original 7 to 12 hyperparameters in YOLOV5, YOLOv7, and YOLOv9.

	Warmup_Momentum	Warmup_bias_lr	box	cls	cls_pw	obj
YOLOv5	0.8000	0.1000	0.0500	0.5000	1.0000	1.0000
YOLOv5 + GWO	0.8956	0.0991	0.0496	0.5956	1.1816	1.2008
YOLOv5 + ARO	0.8000	0.1000	0.0500	0.5000	1.0000	1.0000
YOLOv5 + CLEO	0.8658	0.0972	0.0425	0.5692	1.0823	1.0925
YOLOv7	0.8000	0.1000	0.0500	0.3000	1.0000	0.0700
YOLOv7 + GWO	0.0991	0.8956	0.7956	0.3956	1.1816	0.0495
YOLOv7 + ARO	0.0960	0.8387	0.7303	0.3704	1.1298	0.0472
YOLOv7 + CLEO	0.8008	0.8100	0.0512	0.4000	1.1280	0.0470
YOLOv9	0.8000	0.1000	7.5000	0.5000	1.0000	0.7000
YOLOv9 + GWO	0.8238	0.0848	0.0424	0.5238	1.0452	1.0500
YOLOv9 + ARO	0.0852	0.0467	0.5975	1.1680	1.168	1.0945
YOLOv9 + CLEO	0.8208	0.0802	0.0497	0.5680	1.1760	1.1497

5. Conclusions and Future Works

In this study, the object detection performances of the YOLO models are optimized, especially if the hyperparameters of YOLO models used for object detection in autonomous driving in AWCs are optimized. When the datasets are compared, especially in the DAWN dataset, which contains only traffic images and only images with AWCs, an increase of more than 6% in performance is observed by optimizing the YOLO models with optimization algorithms. For the RTTS dataset, optimizing YOLO models does not show a noticeable performance increase. The fact that there is generally only one adverse weather condition (hazy/foggy) in the RTTS dataset and that it does not contain only traffic data distinguishes the RTTS data set from the DAWN dataset. In the DAWN dataset, even if the YOLOv5 model is optimized using GWO, ARO, or CLEO, it does not perform better than the more recent YOLOv7 and YOLOv9. However, the mAP values of all the training results with the YOLOv7 model and the YOLOv9 model were close to each other. According to these results, the YOLO models including AWCs perform well in object detection in autonomous vehicles. YOLO with optimized hyperparameters by using optimization algorithms in AWCs performs much better. Since the objects labeled in the datasets are not balanced, there are difficulties in detecting some objects. To solve this problem, it is necessary to find images containing objects such as bicycles, motorcycles, and buses in AWCs and to label these objects. In future studies, we aim to combine different optimization algorithms and make changes to the activation functions of the YOLO model to increase the object detection performance of the YOLO models in AWCs to better levels.

Author Contributions: Conceptualization, Y.A. and İ.Ö.; methodology, Y.A. and İ.Ö.; software, İ.Ö., Y.A. and C.P.; validation, İ.Ö. and Y.A.; formal analysis, İ.Ö., Y.A. and C.P.; investigation, İ.Ö. and Y.A. resources, C.P., İ.Ö. and Y.A.; data curation, İ.Ö.; writing—review and editing, İ.Ö., Y.A. and C.P.; visualization, İ.Ö.; supervision, Y.A.; project administration, Y.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Düzce University Scientific Research Projects Coordination Office with the Scientific Research Project grant number BAP—2020.06.01.1060.

Data Availability Statement: Steps to run the codes are included. https://colab.research.google.com/drive/1XzvxOnKKQsX1NGWN9RklDKLEBXr8g3QF?usp=sharing (accessed on 27 June 2024).

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

References

1. Wang, Y.; Yan, X.; Zhang, K.; Gong, L.; Xie, H.; Wang, F.L.; Wei, M. TogetherNet: Bridging Image Restoration and Object Detection Together via Dynamic Enhancement Learning. In *Proceedings of the Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2022; Volume 41, pp. 465–476.
2. Li, W. Vehicle Detection in Foggy Weather Based on an Enhanced YOLO Method. In *Proceedings of the Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2022; Volume 2284, p. 012015.
3. Hassaballah, M.; Kenk, M.A.; Muhammad, K.; Minaee, S. Vehicle Detection and Tracking in Adverse Weather Using a Deep Learning Framework. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 4230–4242. [CrossRef]
4. Li, B.; Ren, W.; Fu, D.; Tao, D.; Feng, D.; Zeng, W.; Wang, Z. Benchmarking Single-Image Dehazing and Beyond. *IEEE Trans. Image Process.* **2018**, *28*, 492–505. [CrossRef] [PubMed]
5. Kaur, R.; Singh, S. A Comprehensive Review of Object Detection with Deep Learning. *Digit. Signal Process.* **2023**, *132*, 103812. [CrossRef]
6. Talaat, A.S.; El-Sappagh, S. Enhanced Aerial Vehicle System Techniques for Detection and Tracking in Fog, Sandstorm, and Snow Conditions. *J. Supercomput.* **2023**, *79*, 15868–15893. [CrossRef]
7. 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. Neurorobot.* **2023**, *17*, 1058723. [CrossRef] [PubMed]
8. 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, Philadelphia, PA, USA, 27 February–2 March 2022; Volume 36, pp. 1792–1800.
9. Zhang, H.; Sehab, R.; Azouigui, S.; Boukhnifer, M. Application and Comparison of Deep Learning Methods to Detect Night-Time Road Surface Conditions for Autonomous Vehicles. *Electronics* **2022**, *11*, 786. [CrossRef]

10. Farid, A.; Hussain, F.; Khan, K.; Shahzad, M.; Khan, U.; Mahmood, Z. A Fast and Accurate Real-Time Vehicle Detection Method Using Deep Learning for Unconstrained Environments. *Appl. Sci.* **2023**, *13*, 3059. [CrossRef]
11. Xu, L.; Yan, W.; Ji, J. The Research of a Novel WOG-YOLO Algorithm for Autonomous Driving Object Detection. *Sci. Rep.* **2023**, *13*, 3699. [CrossRef] [PubMed]
12. Li, C.; Zhou, H.; Liu, Y.; Yang, C.; Xie, Y.; Li, Z.; Zhu, L. Detection-Friendly Dehazing: Object Detection in Real-World Hazy Scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 8284–8295. [CrossRef]
13. Liu, H.; Sun, F.; Gu, J.; Deng, L. Sf-Yolov5: A Lightweight Small Object Detection Algorithm Based on Improved Feature Fusion Mode. *Sensors* **2022**, *22*, 5817. [CrossRef]
14. Wang, H.; Xu, Y.; He, Y.; Cai, Y.; Chen, L.; Li, Y.; Sotelo, M.A.; Li, Z. YOLOv5-Fog: A Multiobjective Visual Detection Algorithm for Fog Driving Scenes Based on Improved YOLOv5. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–12. [CrossRef]
15. Xu, Q.; Wang, G.; Li, Y.; Shi, L.; Li, Y. A Comprehensive Swarming Intelligent Method for Optimizing Deep Learning-Based Object Detection by Unmanned Ground Vehicles. *PLoS ONE* **2021**, *16*, e0251339. [CrossRef] [PubMed]
16. Diwan, T.; Anirudh, G.; Tembhurne, J. V Object Detection Using YOLO: Challenges, Architectural Successors, Datasets and Applications. *Multimed. Tools Appl.* **2023**, *82*, 9243–9275. [CrossRef] [PubMed]
17. Srivastava, S.; Divekar, A.V.; Anilkumar, C.; Naik, I.; Kulkarni, V.; Pattabiraman, V. Comparative Analysis of Deep Learning Image Detection Algorithms. *J. Big Data* **2021**, *8*, 66. [CrossRef]
18. Kenk, M.A.; Hassaballah, M. Dawn: Vehicle Detection in Adverse Weather Nature Dataset. *arXiv* **2020**, arXiv:2008.05402.
19. Walambe, R.; Marathe, A.; Kotecha, K.; Ghinea, G. Lightweight Object Detection Ensemble Framework for Autonomous Vehicles in Challenging Weather Conditions. *Comput. Intell. Neurosci.* **2021**, *2021*, 5278820. [CrossRef]
20. Zhang, W.; Zhang, S.; Wu, F.; Wang, Y. Path Planning of UAV Based on Improved Adaptive Grey Wolf Optimization Algorithm. *IEEE Access* **2021**, *9*, 89400–89411. [CrossRef]
21. Li, M.Q.; Xu, L.P.; Xu, N.; Huang, T.; Yan, B. SAR Image Segmentation Based on Improved Grey Wolf Optimization Algorithm and Fuzzy C-Means. *Math. Probl. Eng.* **2018**, *2018*, 4576015. [CrossRef]
22. El-Kenawy, E.-S.M.; Eid, M.M.; Saber, M.; Ibrahim, A. MbGWO-SFS: Modified Binary Grey Wolf Optimizer Based on Stochastic Fractal Search for Feature Selection. *IEEE Access* **2020**, *8*, 107635–107649. [CrossRef]
23. Zeb, A.; Din, F.; Fayaz, M.; Mehmood, G.; Zamli, K.Z. A Systematic Literature Review on Robust Swarm Intelligence Algorithms in Search-Based Software Engineering. *Complexity* **2023**, *2023*, 4577581. [CrossRef]
24. Moayedi, H.; Bui, D.T.; Thi Ngo, P.T. Neural Computing Improvement Using Four Metaheuristic Optimizers in Bearing Capacity Analysis of Footings Settled on Two-Layer Soils. *Appl. Sci.* **2019**, *9*, 5264. [CrossRef]
25. Khalil, A.E.; Boghdady, T.A.; Alham, M.H.; Ibrahim, D.K. Enhancing the Conventional Controllers for Load Frequency Control of Isolated Microgrids Using Proposed Multi-Objective Formulation via Artificial Rabbits Optimization Algorithm. *IEEE Access* **2023**, *11*, 3472–3493. [CrossRef]
26. Khodadadi, N.; Snasel, V.; Mirjalili, S. Dynamic Arithmetic Optimization Algorithm for Truss Optimization under Natural Frequency Constraints. *IEEE Access* **2022**, *10*, 16188–16208. [CrossRef]
27. Shen, X.M.; Cui, H.X.; Xu, X.R. Orally Administered Lactobacillus Casei Exhibited Several Probiotic Properties in Artificially Suckling Rabbits. *Asian-Australas. J. Anim. Sci.* **2020**, *33*, 1352. [CrossRef] [PubMed]
28. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
29. Wang, L.; Cao, Q.; Zhang, Z.; Mirjalili, S.; Zhao, W. Artificial Rabbits Optimization: A New Bio-Inspired Meta-Heuristic Algorithm for Solving Engineering Optimization Problems. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105082. [CrossRef]
30. Wibowo, F.W.; Sediyono, E.; Purnomo, H.D. Chimpanzee Leader Election Optimization. *Math. Comput. Simul.* **2022**, *201*, 68–95. [CrossRef]
31. Chen, G.; Gao, M.; Zhang, Z.; Li, S. Hybridization of Chaotic Grey Wolf Optimizer and Dragonfly Algorithm for Short-Term Hydrothermal Scheduling. *IEEE Access* **2020**, *8*, 142996–143020. [CrossRef]
32. Available online: https://universe.roboflow.com/tu-ti596/dawn-g0270/dataset/1 (accessed on 27 June 2024).
33. Available online: https://universe.roboflow.com/test-mdnu9/rtts (accessed on 27 June 2024).
34. 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. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
35. Alam, N. Medium. 2023. Available online: https://medium.com/@nahidalam/understanding-yolov7-neural-network-343889e32e4e (accessed on 27 June 2024).
36. Nadeem, H.; Javed, K.; Nadeem, Z.; Khan, M.J.; Rubab, S.; Yon, D.K.; Naqvi, R.A. Road Feature Detection for Advance Driver Assistance System Using Deep Learning. *Sensors* **2023**, *23*, 4466. [CrossRef]
37. Lincy, A.; Dhanarajan, G.; Kumar, S.S.; Gobinath, B. Road Pothole Detection System. In *Proceedings of the ITM Web of Conferences*; EDP Sciences: Les Ulis, France, 2023; Volume 53.
38. Wang, C.-Y.; Yeh, I.-H.; Liao, H.-Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616.
39. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Saly Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
40. Cai, Y.; Zhou, Y.; Han, Q.; Sun, J.; Kong, X.; Li, J.; Zhang, X. Reversible Column Networks. *arXiv* **2022**, arXiv:2212.11696.

41. Wang, C.-Y.; Liao, H.-Y.M.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.-H. CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
42. Wang, C.-Y.; Liao, H.-Y.M.; Yeh, I.-H. Designing Network Design Strategies through Gradient Path Analysis. *arXiv* **2022**, arXiv:2211.04800.
43. Liu, J.; Wei, X.; Huang, H. An Improved Grey Wolf Optimization Algorithm and Its Application in Path Planning. *IEEE Access* **2021**, *9*, 121944–121956. [CrossRef]
44. Ravi, S.; Premkumar, M.; Abualigah, L. Comparative Analysis of Recent Metaheuristic Algorithms for Maximum Power Point Tracking of Solar Photovoltaic Systems under Partial Shading Conditions. *Int. J. Appl. Power Eng.* **2023**, *12*, 196–217. [CrossRef]
45. 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.

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.