Article

Deep Camera-Radar Fusion with Attention Framework for Autonomous Vehicle Vision in Foggy Weather Conditions

Isaac Ogunrinde [1,*], Shonda Bernadin [1]

[1] Department of Electrical and Computer Engineering, FAMU-FSU College of Engineering, Tallahassee, FL 32310; bernadin@eng.famu.fsu.edu.

* Correspondence: isaac1.ogunrinde@famu.edu;

Abstract: AVs suffer reduced maneuverability and performance due to the degradation in sensor performances in fog. Such degradation causes significant object detection errors essential for AVs' safety-critical conditions. For instance, YOLOv5 performs significantly well under favorable weather but suffers miss detections and false positives due to atmospheric scattering caused by fog particles. Existing deep object detection techniques often exhibit a high degree of accuracy. The drawback is being sluggish at object detection in fog. Object detection methods with fast detection speed have been obtained using deep learning at the expense of accuracy. The problem of the lack of balance between detection speed and accuracy in fog persist. This paper presents an improved YOLOv5-based multi-sensor fusion network that combines radar's object detection with a camera image bounding box. We transformed radar detection by mapping the radar detections into a two-dimensional image coordinate and projected the resultant radar image on the camera image. Using the attention mechanism, we emphasized and improved important feature representation used for object detection while reducing high-level feature information loss. We trained and tested our multi-sensor fusion network on clear and multi-fog weather datasets obtained from the CARLA simulator. Our result shows that the proposed method significantly enhances the detection of distant and small objects. Our small CR-YOLOnet model best strikes a balance between accuracy and speed with an accuracy of 0.849 at 69 fps.

Keywords: Sensor fusion; object detection; deep learning; autonomous vehicles; camera-radar; adverse weather; fog; attention module

1. Introduction

AVs encounter several difficulties under adverse weather conditions, such as snow, fog, haze, shadow, and rain [1-8]. AVs may suffer from poor decision-making and control if their perception systems are degraded by adverse weather. When water vapor condenses in the sky, it obscures the view of the surrounding area, resulting in fog. Fog can make driving unsafe because it obscures visibility. The signal-to-noise ratio (SNR) is reduced while measurement noise rises dramatically under foggy conditions. Unsafe behavior and road accidents might be caused by sensor data that is too noisy.

Machine vision in fog can fall as low as 1000 meters in moderate fog and as low as 50 meters in heavy fog [9,10]. Camera sensors are one of the significant sensors used for object detection because of their low cost and the large number of features they provide [11]. Under fog, the camera's performance is limited due to visibility degradation. The quality of the image taken by a camera system can be substantially distorted by fog. In fog, Lidar suffers reflectivity degradation and reduction in distance measured. However, radars tend to perform better than cameras and lidars under adverse weather since radars are unaffected by changes in environmental conditions [11,12]. Radars employ the Doppler effect to determine the distance and velocity of objects by monitoring the reflection of radio waves. When it comes to object classification, radars fall short. Because radars can only detect objects, they cannot classify what kind of object they are detecting since radar detections are far too sparse [13,14]. The sparse nature of radar point clouds collected with many vehicular radars (usually 64 points or less) might explain this [15].

AVs are often outfitted with numerous complementing sensors to provide complementary information that helps attain the necessary accuracy when combined. Multi-sensor fusion combines data from numerous sensors to achieve higher object detection/classification accuracy and performance than with a single sensor modal system [11]. Therefore, an essential subject for AVs is the combination of radar and other sensors, such as cameras. Radar-camera fusion systems can offer useful depth information for all observed objects in an autonomous driving situation. Radar sensors construct detections of nearby objects for subsequent usage, while the bounding boxes on camera data can be used to verify and validate prior radar detections using deep learning-based object detection methods [14].

There have been significant contributions in object detection and classification using deep learning. Besides AVs technology, object detection has found application in other fields, including surveillance and security [16], medicine [17], robotics [18,19], military [20], etc. In [21], deep Convolutional Neural Networks (CNN) was first utilized for image classification in 2012. However, when it comes to vehicular radars, it's not uncommon for parts of the observations to have incomplete, distorted, and poor-quality data. Beam obstruction, instrument malfunction, blind spots, close-to-the-ground mounting, inclement weather such as fog, and many more contribute to these problems. Images obtained by camera consist of color and feature information. This feature information can be used for label classification in an object detection task. The occurrence of fog can drastically distort the feature information of an image due atmospheric scattering and attenuation. These radar and camera problems usually led to inaccuracies in the real time detection of the bounding box of an object or location in an image, especially when the object is not nearby or when the object is too small under medium and heavy foggy weather conditions. Thus, applying single-sensor modal CNN-based object detection algorithms on such distorted data has proven inefficient [1,2].

YOLOv5 [22], a state-of-the-art object detection algorithm suffer from miss detections and false positives due to atmospheric scattering caused by fog particles. Existing deep learning-based object detection techniques that exhibit a high degree of accuracy have slow object detection speed in foggy weather conditions. However, several deep learning-based object detection methods achieved fast detection speed at the expense of accuracy. Therefore, the problem of the lack of balance between detection speed and accuracy in fog weather application persist. The uniqueness of radar signals and the scarcity of publicly available datasets [23] containing both camera and radar datasets [24-29] under foggy weather conditions have resulted in a limitation of AV research in this area. Very few datasets, such as [27] and [29], are available for AV research that include camera and radar information under foggy weather conditions. To accommodate the needs of AVs in terms of previously mentioned problems as related to AVs' environmental perception in fog, we made the following contributions:

- Using the image data, we demonstrate that sensor measurements are severely impacted by atmospheric distortion in foggy conditions.
- We present a deep learning-based camera-radar fusion network (CR-YOLOnet) using YOLOv5 [22] as the baseline for object detection, as shown in Figure 1. We have made the following improvements to the baseline YOLOv5 to achieve CR-YOLOnet. (i) The CR-YOLOnet can take input from camera and radar sources compared to the single modal system in the baseline YOLOv5. There are two CSPDarknet [30] backbone networks with which CR-YOLOnet extracts feature maps, each for the camera and radar channels. (ii) Using two connections inspired by the concepts of residual networks, the feature information from the backbone network was sent to the feature fusion layers. The two connections' purpose is to improve the backpropagation of gradient in our network while minimizing feature information loss for relatively tiny objects in fog. (iii) We enhanced CR-YOLOnet with an attention framework to detect multi-scale

object sizes in foggy weather conditions. To draw more emphasis and improve the feature representation of the features that helps with object detection, attention modules are incorporated into the fusion layers. The attention module also helps to address the issue of high-level feature information loss to boost the detector performance.

- We simulated an autonomous driving environment on CARLA [31] simulator, from which we collected both camera and radar data. We make use of both clear and foggy weather conditions for CR-YOLOnet training and test evaluations. To further evaluate CR-YOLOnet, we compared the matching size of the baseline YOLOv5 to the small, medium, and large models of our CR-YOLOnet.

This paper's remaining sections are structured as follows: we discussed related works in section 2, we presented our methodology in section 3, we presented and discussed our results in section 4, and section 5 consists of the conclusion.

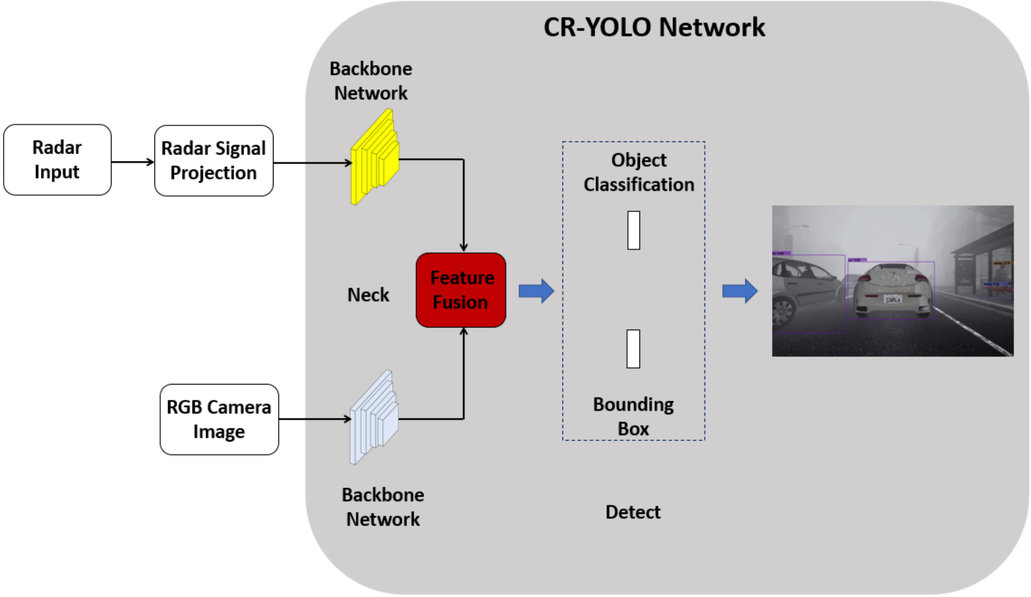


Figure 1: The proposed camera-radar fusion network (CR-YOLOnet).

2. Related Works

2.1. Object Detection by Camera Only

Object detection can help identify and determine each object instance's spatial size and position in an image if instances of previously defined object categories exist in the image [32]. Usually, object detection algorithms generate many potential region proposals from which the best feasible candidates are selected [33]. The two categories of CNN-based object detection techniques are [11]: (i) two-stage object detectors and (ii) one-stage object detectors. Girshick et al. in [34] proposed the first CNN-based object detection algorithm. R-CNN [34], Fast R-CNN [35], and Faster R-CNN [36] are examples of two-stage object detection algorithms. The two-stage detectors isolate the task of localizing objects using regions of interest from the task of classifying objects.

Redmon et al. [37-39] proposed YOLO, a one-stage detector. YOLO and its derivatives can instantly predict bounding boxes and the object class after extracting features from an input image. One-stage object detectors generate candidate regions, instantly used to classify and predict the target's spatial location [1]. Backbone networks such as feature pyramid networks (FPNs) [40] together with one-stage detectors such as YOLO [37], YOLO9000 [38], YOLOv3 [39], or SSD [41] were used to detect objects via numerous detection branches in one operation instead of predicting the potential locations and classifying them later. Because one-stage detectors do not depend on RPN for predicting potential regions thus, they are more efficient than two-stage detectors and are highly used for real-time object detection applications [33].

Several methods have been proposed in the literature to address autonomous driving in adverse weather conditions using the camera. Walambe et al. [42] proposed an ensemble-based method to enhance AV's ability to detect objects such as vehicles and pedestrians under challenging settings, such as inclement weather. Multiple deep-learning models were ensembled with alternative voting techniques for object detection while using data augmentation to improve the models' performance. In [43], Gruber et al. suggested that backscatter may be significantly reduced with gated imaging, making it a viable solution for Avs operating in severe weather conditions. In addition to offering intensity images, gated images may produce properly aligned depth data. However, eye safety standards prevent the illumination from getting beyond the sunshine, making it impossible for gated imaging to work well on extremely bright days.

Tumas et al. [44] introduced 16-bit depth modifications into YOLOv3 algorithms for pedestrian detection in severe weather conditions. While the authors employed an onboard precipitation sensor to adjust image intensity, they could not implement a real-time image enhancement for annotations collected in rain or fog. In [45], Sommer et al. used the RefineDet detection framework that consists of some Faster R-CNN and SSD detection frameworks for Vehicle Detection in Traffic Surveillance Data. To achieve a robust detection capability, the authors proposed an ensemble network that combined two detectors, namely SENets and ResNet-50, as the base network. However, the authors only focused on night-time and rainy scenarios. Sindagi et al. [46] proposed an unsupervised prior-based domain adaptive object detection framework for hazy and rainy conditions based on the Faster-RCNN framework. The authors trained an adaptation process using a prior-adversarial loss network to generate weather-invariant features by diminishing the weather-related data in the features. However, some improvement is required on the prior-adversarial loss network. In 2020, Hamzeh et al. [4] developed a quantitative measure of the effect of raindrop distortion on the performance of deep learning-based object detectors algorithms (including Faster R-CNN, SDD, and YOLOv3) based on the comparison between raindrop-distorted (wet) images and clear images. With the proposed quantitative measure, the amount of degradation that occurs in the detection performance of an object detector can be predicted. Liu et al. [2] conducted a study that analyzed how perception in foggy conditions impacts the detection recall using a single modal approach based on camera images. Camera images collected were characterized by deploying a Faster RCNN approach for object detection. Experimental results in [2] show that detection recall is less than 57.75% in heavy fog conditions. This implies that a single modal system, such as camera-only architecture, is insufficient to handle target detection issues under adverse weather conditions.

Bochkovskiy et al. [30] proposed YOLOv4 with CSPDarknet53 backbone and CIOU LOSS for evaluating prediction boxes. Jocher et al. [22] proposed YOLOv5 that uses the CSPDarknet53 backbone, the architecture of the feature pyramid network (FPN) [40], and the pixel aggregation network (PAN) [47] as the neck. YOLOv5, with large model size, tends to have higher accuracy but low detection speed. The performance of YOLOv5 with a small model size is similar to YOLOv4 in terms of accuracy but faster than YOLOv4 in the speed of detection. As a result, the YOLOv5 network will serve as the baseline of our research and improvement in this study.

2.2 Object Detection by Fused Camera and Radar Sensors

Radar signals and camera data have lately been combined using neural networks to accomplish various AV tasks. Radar signal representation methods include radar occupancy grid maps, radar signal projection, radar point clouds, micro-Doppler signature, Range-Doppler-Azimuth tensor, etc. [11]. Several radar signal processing approaches in the literature include occupancy grid maps [48], range-velocity-azimuth maps [49], and radar point clouds [50]. As a result, several researchers [11] have suggested numerous alternative ways to represent radar signals in deep learning.

Our focus in this work is the radar signal projection method. Transforming radar signals such as point clouds or detections into a two-dimensional image coordinate or a three-dimensional bird's eye perspective is a technique known as radar signal projections. The radar, camera, and target coordinates contribute significantly to this scenario. The intrinsic and extrinsic camera calibrating matrices are used to execute the radar point cloud transformation. The resulting radar images overlayed on the image grid. The radar image includes the radar detections and their properties which can be fed into the DCNN network. In the literature, multiple deep learning-based fusion methods based on vision and radar signals have been proposed.

Nabati et al. [51] suggested a technique based on a radar region proposal for object detection. Using the method in [51], the two-stage object detectors were eliminated, which imposed a heavy strain on region proposal creation. Radar detections were mapped into an image plane, and the resulting image contains object proposals and anchor boxes. This approach uses radar detection instead of vision to acquire region suggestions, which saves time and effort while providing better detection results. Radar and vision sensors were combined by Chadwick et al. [52] to detect objects in the distance more precisely. It was first necessary to generate two extra imaging streams based on range and radial velocity to provide a format of the radar images in an image plane. A concatenation approach combined the radar and vision feature representations obtained from an SSD model.

Nobis et al. [53] introduced a neural network-based object detection approach by projecting sparse radar signals onto an image vertical plane. The network was able to automatically determine the optimal level of sensor data to use to increase detection accuracy. Black-in, a novel training method that prioritizes the use of a certain sensor at a specific period to obtain better outcomes, was also introduced. Meyer et al. [54] used DCNNs to perform a low-level combination of radar point clouds and camera images to detect 3D objects. The DCNNs learn to recognize vehicles using camera images and bird eye view images generated from radar point clouds and surpass Lidar camera-based systems when tested. Zhang et al. [55] proposed a radar and vision fusion system to detect and identify objects in real-time. First, the object's position and velocity were detected and obtained using radar. Subsequently, the radar data is then projected into the corresponding image plane. A deep learning system then uses ROI for target detection and identification.

John et al. [56] used the Yolo object detector to combine the separate data acquired from radar and monocular camera sensors to identify obstacles in inclement weather better. Using two input channels, feature vectors were extracted, one for the camera and the other for the radar. Two output channels were used to categorize targets into groups of smaller and larger objects. A sparse radar image was created by projecting radar point clouds onto the image plane. Aiming to lessen the computational load relating to real-time applications, John et al. [57] suggested a multitask learning framework based on the deep fusion of radar and camera information for joint semantic segmentation and obstacle identification.

Zhou et al. [58] proposed an object detection system that was based on the deep fusion of information from mm-wave radar and cameras utilizing YOLO detector. The data from the radar was utilized to generate a radar image using a single channel. The radar image is combined with the RGB camera image to produce a four-channel image subsequently fed into the YOLO object detector. To enhance the detection of small and minimally predictable objects, Chang et al. proposed a spatial attention module to be used in conjunction with millimeter-wave radar and vision fusion target detection algorithms [59]. None of these previous studies emphasized detecting small and/or remotely distant under medium and heavy fog conditions.

3. Method

3.1. Sensor Calibration and Coordinate Transformation

doi:10.20944/preprints202305.2180.v1

Measurement errors grow with distance since radar and image are often mounted in different locations on the ego vehicle. As a result, the shared observation region between the camera and radar requires a combined calibration effort. The vehicle's motion defines a local right-hand coordinate system in the ego vehicle coordinate system. The local coordinate system is conveyed with the vehicle as it travels through the environment. The x-axis indicates the path of motion, whereas the y-axis is parallel to the front axle, which serves as the starting point. The camera models employ three-dimensional coordinates. For example, the camera's x-y-z coordinate system has its origins in the camera's viewpoint. Images acquired by camera sensors employ an image coordinate frame (x, y) and pixel coordinate (u, v) as reference points for composition. The coordinate system of choice is polar coordinates when it comes to radar detection. The detected objects are referenced using polar coordinates. Thus, a target may be recorded as an x-y coordinate system vector in a vector space. The canonical coordinate system for radar comprises three elements: an azimuth α, a distance between the object r, in the direction of the sensor origin. By measuring point P distance and its azimuth from the radar, we can estimate where P is in the world coordinate system [60,61].

The observations of the camera and radar detections can be associated using the information in a shared world coordinate system given by $[X; Y; Z; 1]$. The camera calibration parameters can be used to project the radar detections onto the camera's coordinate system and the image plane given by $[x; y; 1]$. The calibration parameters of the camera can break them down into two matrices: intrinsic and extrinsic. With the intrinsic parameters matrix given in terms of as [60,61]:

$$C = \begin{bmatrix} f_x & 0 & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{1}$$

where $f_x = f/dx$ and $f_y = f/d_y$ Such that f represents the focal length of the camera, d_x and d_y represents the physical dimensions of an individual pixel in the x-y axes directions, respectively, f_x and f_y represents the scale factors on u and v axes, u_0 and v_0 represents the central point offset of the camera.

The extrinsic camera parameter can be expressed as:

$$\begin{bmatrix} R_{3\times3} & T_{3\times1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{21} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

where R represents the rotation parameter matrix and T is the translation parameter matrix used for mapping radar detection point to projection point P coordinates in the image plane. Thus, the radar detections may be mapped to their equivalent visual representations. After the mapping, the detections that fall outside the image frame are disregarded to ensure accuracy. The coordinate mapping from the world coordinate system to the image plane of image coordinate systems is as follows:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = C \begin{bmatrix} R_{3\times3} & T_{3\times1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{3}$$

where x and y represent the projection point P coordinates in the image plane.

3.2. Fog Imaging Model

doi:10.20944/preprints202305.2180.v1

Physical atmospheric scattering models are shown in Figure 2. The attenuation factor, transmission model, and airlight model make up the physical atmospheric scattering model. Atmospheric scattering reduces the amount of light that may be absorbed for imaging under foggy conditions. Therefore, the target image's object textures and edge features may be diminished. Attenuation and interference occur before the reflected light reaches the camera in foggy weather. An airtight concept allows light rays to be scattered before they reach the imaging camera. Instead of being a scene light from the item in the photograph, transmitted lights include fog elements that obscure the images.

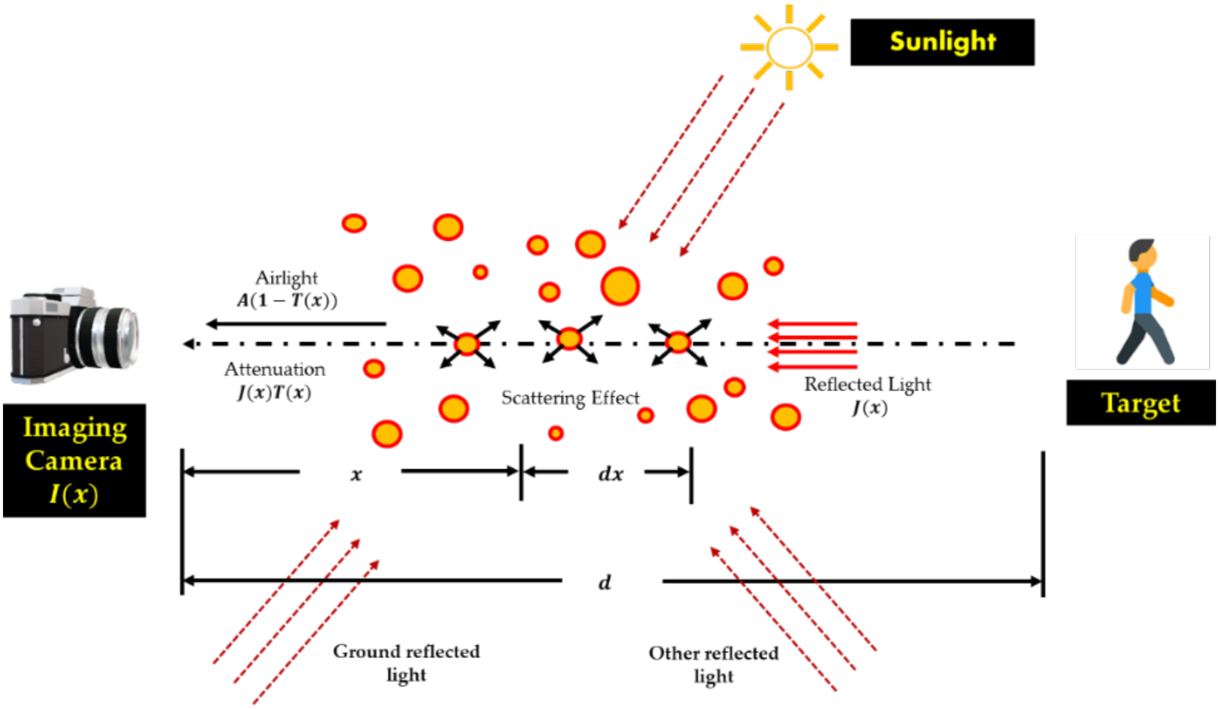


Figure 2: An atmospheric scattering phenomenon of foggy imaging model

An image model suggested by Koschmieder [62] has been frequently used in the scientific literature [1]:

$$I(x) = J(x)t(x) + A[1 - t(x)], \tag{1}$$

$I(x)$ denotes the picture captured by the camera, $J(x)$ indicates the scene radiance image, $t(x)$ denotes the transmission map, and A denotes the airlight vector, which is homogenous for each pixel in the image. The attenuation factor is represented by $J(x)t(x)$, while the atmospheric components are represented by $A[1 - t(x)]$. The undetermined parameters of a hazy single input picture I are represented by the letters A, t, and J. To acquire the restored picture (recovered image) $\hat{J}$, the amount of ambient light $\hat{A}$ and transmission $\hat{t}$ can be determined using equation (2).

$$\hat{J}(x) = \hat{I}\,(x) - \frac{\hat{A}[1-\hat{t}(x)]}{\hat{t}(x)}, \tag{2}$$

According to Narasimhan et al. [63], the visual imaging model of a foggy scenario can be regarded as the outcome of concatenating the attenuation and interference models, as shown in Figure 2. As a result of both attenuation and interference, fog can seriously degrade the quality of the image being captured in a machine. The theoretical model of the visual imaging model of a foggy scenario can also be represented as follows [2]:

$$E(d,z) = E_0(z)e^{-\beta(z)} + E_\infty(z)(1 - e^{-\beta(z)}), \tag{3}$$

where $E_0(z)e^{-\beta(z)}$ represents the attenuation model, $E_\infty(z)(1 - e^{-\beta(z)})$ represents the interference model, light waves have a certain wavelength z, the atmospheric scattering coefficient is denoted by $E_0(z)$ and it measures the light's capacity to disperse per unit volume, the depth of the scene be represented by d, the scattering coefficient is denoted by β, $\beta(z)$ indicates the intensity of the target obstacle's light as it scatters through the atmosphere and reaches the camera.

As mentioned earlier, the scattering impact of incoming light on airborne particles in the atmosphere will reduce the intensity of light that ultimately reaches the camera [11]. We consider the relationship between depth d of the scene and transmission t. Also, we conder the effect of image degradation due to attenuation on the visibility of the image. Consider an observer (imaging camera) at distance d(x) from a scene point at position x. The relationship between the transmission t and depth d is expressed as follows in equation (2.3) [64]:

$$t(x) = exp\left(-\int_0^{d(X)} \beta(z)dz\right), \tag{4}$$

where $\int_0^{d(x)}$ is the distance between the imaging camera and the scene point at x, and β represents the atmospheric scattering coefficient. If the atmosphere exhibits homogeneous physical properties, then the scattering coefficient β is the spatial constant. Therefore, equation (4) can be rewritten as:

$$t(x) = e^{-\beta d(x)}, \tag{5}$$

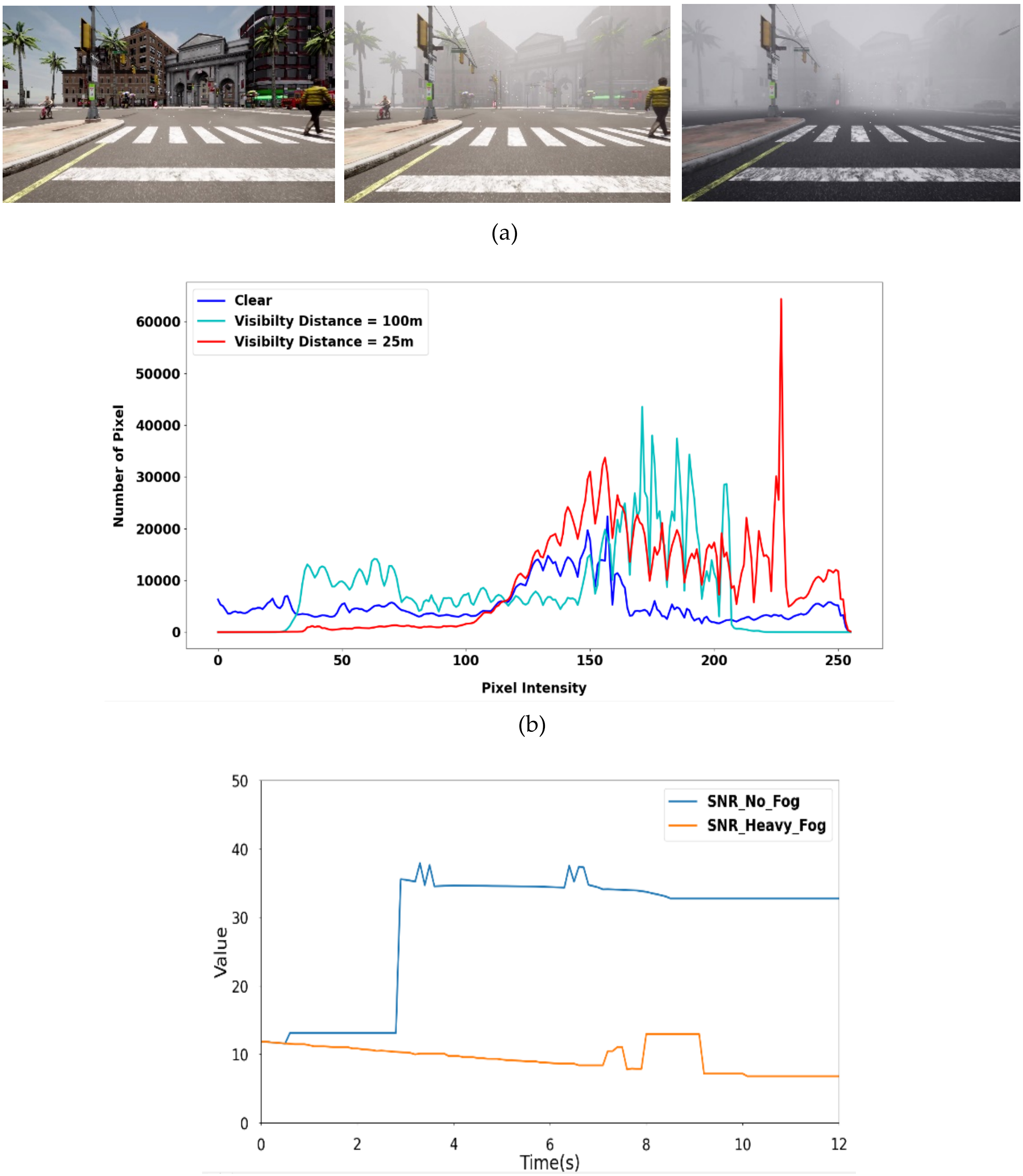
The transmission $t(x) = e^{-\beta d(x)}$ illustrates the unscattered part of the light that gets to the camera. From equation (2.4), we can express $d(x)$ as follows:

$$d(x) = -\frac{lnt(x)}{\beta}, \tag{6}$$

Equation (2.5) implies that the depth can be calculated up to an unknown scale if the transmission can be estimated [64]. The visibility distance, measured in meters, is the maximum distance at which a white and black object loses their distinct contrast. As distance increases in fog, a white and black object seems to be a uniform gray to the human eye. Therefore, the standard maximum contrast ratio is 5% [65].

Figure 3(a) depicts clear, foggy images collected from a real-time autonomous driving simulation at 100m and 25m visibility distances. Figure 3b illustrates the contrast between the grayscale of clear and foggy images at a visibility distance of 100m and 25m. The information about an image's colors and features can be revealed significantly when the image is converted to grayscale. The information about the image's features can be extracted and used for classification purposes in an object detection task. As shown in Figure 3(b), the range of the grayscale of the clear-day image is from around 0 to 250. The grayscale of the foggy images at 100m and 25m visibility distance is highly concentrated between 30 and 210, 100 and 250, respectively. As a result, the detection of objects can be negatively affected by fog because it drastically distorts the image's feature information. [3].

Figure 3(c) shows a simulation of a real-time autonomous driving scene that lasted for 12 seconds in clear (no fog) and at heavy fog conditions with a visibility distance of 25m. Because sensor measurement noise tends to increase significantly in fog, the signal-to-noise ratio (SNR) value decreases dramatically. Figure 3(c) illustrates a higher SNR value in the no-fog scene and a much lower SNR value in the heavy-fog scene.



(a)

(b)

(c)

Figure 3: (a) clear and foggy images at 100m and 25m visibility distance, (b) the comparison of gray scale for clear and foggy images at 100m and 25m visibility distance (c) the comparison of SNR values for clear (no fog) and visibility distance of 25m (heavy fog).

3.3. The Baseline YOLOv5 Model

YOLO is a cutting-edge real-time object detection algorithm, and Yolov5 [22] is built on earlier versions of the YOLO algorithm. YOLO is one of the most effective object detection methods available, with a notable performance on the state-of-the-art results on datasets such as Microsoft COCO [66] and Pascal VOC [67].

The backbone, neck, and head sections are the three fundamental components of the baseline YOLOv5 network as shown in Figure 4. The functionality of the backbone section involves extracting relevant feature data from input images. The neck combines the collected features to create three different scales of feature maps used by the head to detect objects in the image. The YOLOv5 backbone network is CSPDarknet, and the neck consists of the FPN (Feature Pyramid Networks) structure and PAN structure (Spatial Pyramid Pooling) structure.

(i) Backbone:

In Yolov5, Darknet [68] was merged with a cross-stage partial network (CSPNet) [69], resulting in CSPDarknet. The CSPDarknet is made up of convolutional neural networks that use numerous iterations of convolution and pooling to generate feature maps of varying sizes from the input image. As a solution to the issues caused by the repetition of gradient information in large-scale backbones, CSPNet incorporates the gradient transitions into the feature map.

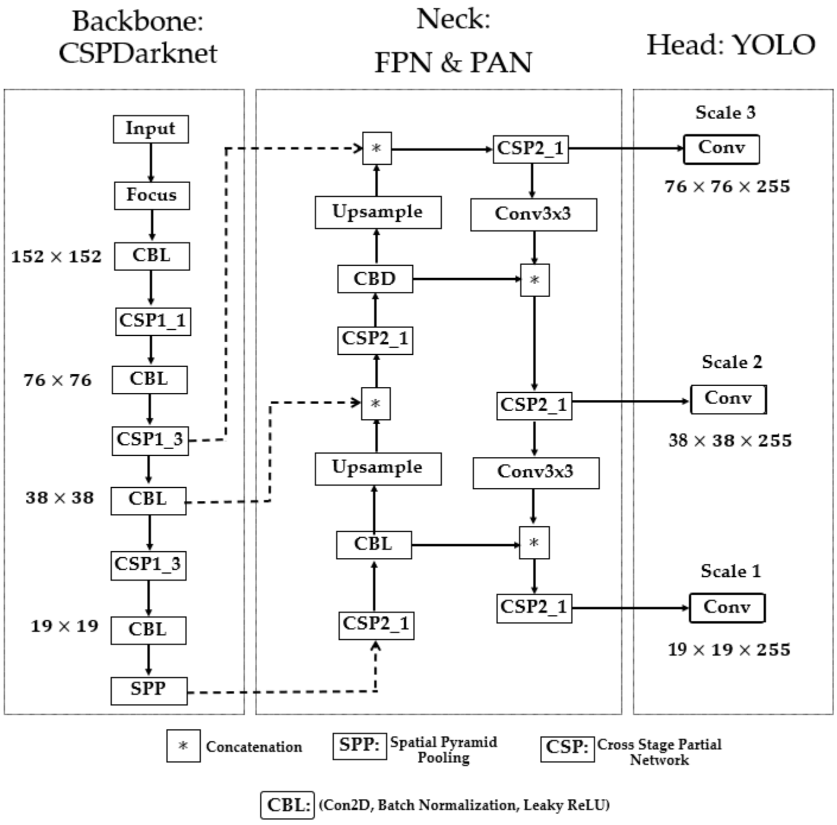


Figure 4: The baseline YOLOv5 architecture.

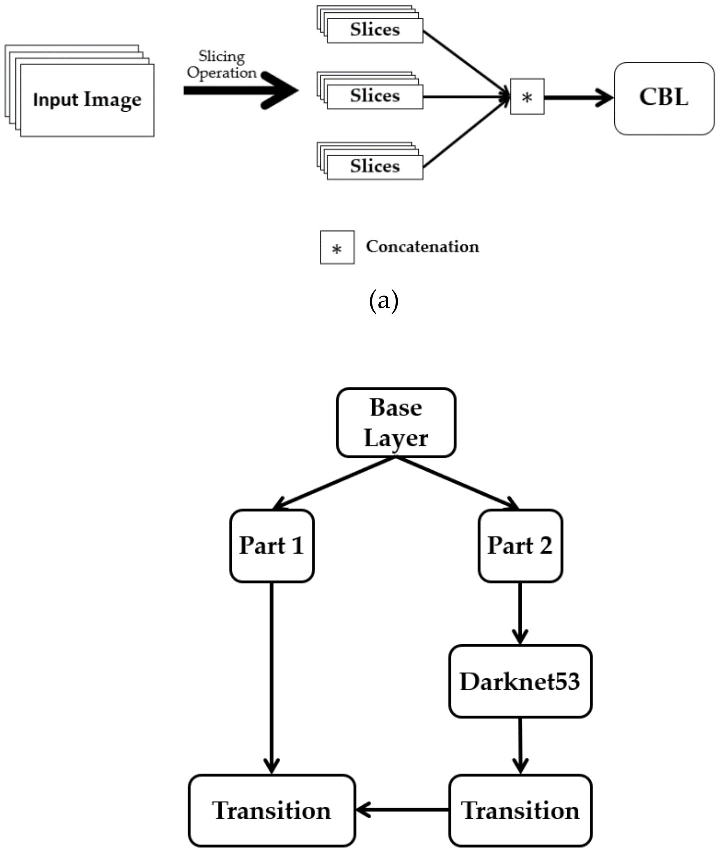
Thus, reducing the model's size, the number of parameters and floating-point operations per second guarantees fast and accurate inference. For the object detection task in fog, it is crucial to have a compact model size, fast detection speed, and high accuracy. The backbone generates four distinct levels of feature maps, including 152 × 152 pixels, 76 × 76 pixels, 38 × 38 pixels, and 19 × 19 pixels.

The backbone focus module (Figure 5a) is used for slicing operations. The purpose of the focus is to improve feature extraction during downsampling. Convolution, batch normalization, and the leaky ReLU Activation Function (AF) are all sub-modules of the CBL module. YOLOv5 implements two distinct cross-stage partial networks (CSP) shown in Figure 5b. Each has a specific function; one is for the neck of a network, and the other is for the backbone. The CSP network uses cross-layer communication between the front and back layers to shrink the model size while preserving accuracy and increasing inference speed. The feature map of the base layer is divided into two distinct parts: the main component and a skip connection. These two parts are then joined using transition, concatenation, and transition to reduce the amount of duplicate gradient information as effectively as possible. Regarding CSP networks, the difference between the backbone and the neck is that the latter uses CBL modules instead of residual units.

Maximum pooling with varying kernel sizes is carried out using the spatial pyramid pooling SPP module [70], shown in Figure 5c. The features are fused through concatenation. The SPP module undertakes dimensionality reduction procedures to convey image features at a higher degree of abstraction. Pooling reduces the feature map's size and the network's computational cost while extracting the essential features.

(ii) Neck:

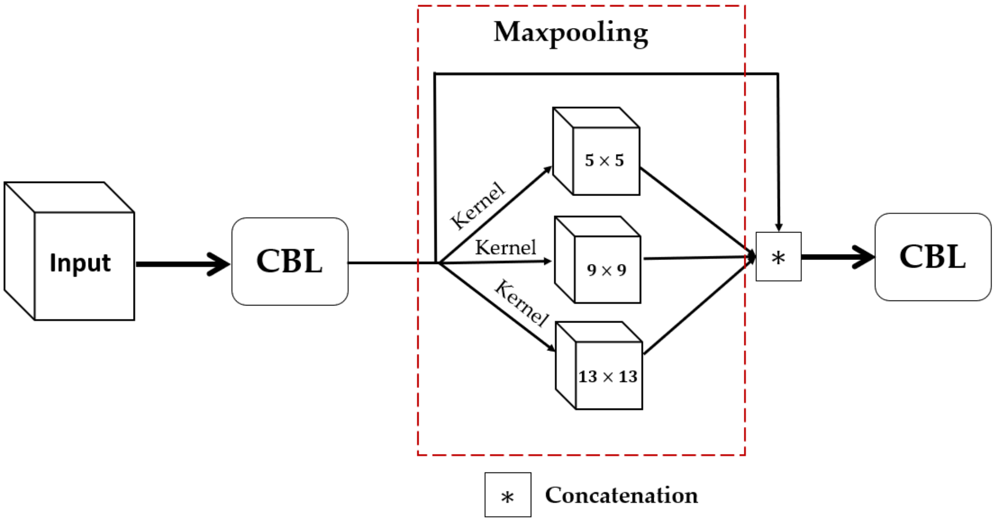
The feature maps from each level are fused by the neck (FPN and PAN) network to learn more contextual information and lessen the amount of data lost in the process. The low-level structures present in the feature maps near the image layer render them ineffective for precise object detection. Feature Pyramid Network (FPN) was designed to extract features to maximize detection speed and accuracy. FPN enables a top-down mechanism to generate higher resolution layers from significant robust semantic feature layers. The PAN architecture effectively transfers localization features in a down-top mechanism from lower to higher feature maps to improve the position accuracy of objects in the image. Thus, feature maps are generated at three different scales at three feature fusion layers.



(a)

(b)

doi:10.20944/preprints202305.2180.v1



(c)

Figure 5: The illustration of (a) the focus architecture, (b) of CSPDarkNet53 architecture, (c) the SPP architecture.

The low-level structures present in the feature maps near the image layer render them ineffective for precise object detection. Feature Pyramid Network (FPN) was designed to extract features to maximize detection speed and accuracy. FPN enables a top-down mechanism to generate higher resolution layers from significant robust semantic feature layers. The PAN architecture effectively transfers localization features in a down-top mechanism from lower to higher feature maps to improve the position accuracy of objects in the image. Thus, feature maps are generated at three different scales at three feature fusion layers.

(iii) Detection Head:

The detection head consists of convolution blocks that take the three different scales of feature maps from the neck layer. With convolution, the detection head yields three distinct sets of detections with resolution levels of 76 × 76 × 255, 38 × 38 × 255, and 19 × 19 × 255. Every grid unit in a feature map correlates to a bigger portion of the original image as the feature map's resolution decreases. This implies that the 76 × 76 × 255 and 19 x 19 x 255 feature maps can adequately detect smaller and large objects.

3.4. Attention Mechanism

When deep CNN reaches a particular depth, numerous studies discovered that it degenerates [71]. Studies have shown that networks' performance does not necessarily improve significantly with depth but can substantially increase computational cost throughout the training phase [72]. Therefore, the attention mechanism was created to train networks to prioritize and devote more focus to relevant feature information while down-ranking those that are less relevant [73]. The attention mechanism informs CNNs where to focus attention and improves the feature representational power of the features that helps with object detection tasks. The human eye provides proof that attention mechanisms are crucial for collecting relevant data [74]. This behavior prompted several studies [74-78] to improve convolutional neural networks' efficiency in image classification problems by including an attention mechanism. In 2018, Woo et al. [76] proposed Convolutional Block Attention Module (CBAM) that integrates spatial and channel attention into a single lightweight mechanism. A considerable performance boost may be achieved with ECA-Net [78], proposed by Wang et al. in 2020. ECA-Net is an efficient channel attention mechanism that can collect information regarding cross-channel relationships.

CBAM [76] was designed to simultaneously capture both channel and spatial attention modules. Since channels of feature maps are treated as feature detectors, the channel attention module focuses on the most important features in input images. This makes the channel attention module an essential application for an image processing task such as object detection in fog. Average-pooling and max-pooling were employed for aggregating the spatial information of the input feature to obtain average-pooled and max-pooled features. For an input feature map $F \in R^{(C \times H \times W)}$, individual channel weights are estimated. Such that the number of channels is C, and the length and width of the feature map in pixels are H and W respectively. The weighted multiplication of channels is useful for drawing more focus to the primary channel features. A shared network (multi-layer perceptron) with one hidden layer is used on both the average-pooling and max-pooling feature descriptors. The element-wise summation of the output vector of both descriptors then generates the channel attention weight map $M_c \in R^{(C \times 1 \times 1)}$ using equation 7. The channel-refined feature maps are obtained by an element-wise multiplication of the original feature map and $M_c \in R^{(C \times 1 \times 1)}$.

$$\begin{aligned} M_c(\boldsymbol{F}) &= \sigma\Big(MLP\big(AvgPool(\boldsymbol{F})\big) + MLP\big(MaxPool(\boldsymbol{F})\big)\Big) \\ &= \sigma\Big(W_1\Big(W_0\big(\boldsymbol{F}^{\boldsymbol{c}}_{\boldsymbol{avg}}\big)\Big) + W_1\big(W_0(\boldsymbol{F}^{\boldsymbol{c}}_{\boldsymbol{max}})\big)\Big), \end{aligned} \tag{7}$$

where σ is the sigmoid activation function, W_1 and W_0 are the multi-layer perceptron weights, $\boldsymbol{F}^{\boldsymbol{c}}_{\boldsymbol{avg}}$ is the average-pooled features and $\boldsymbol{F}^{\boldsymbol{c}}_{\boldsymbol{max}}$ is the max-pooled features.

Next, the spatial component uses the channel refined features from the channel submodule to generate a 2D spatial attention map. The element-wise multiplication of the spatial attention weight map and the input channel attention feature map generates the final refined feature map from the attention mechanism [79]. The spatial attention module pays the most attention to the object's position in the image frame. This is achieved by combining the spatial features in individual space using the weighted sum of spatial features. The overall refined features are obtained by multiplying the channel refined features by the 2D spatial attention map. For a channel-refined feature map $F_c \in R^{(C \times H \times W)}$, the convolution of the average-pooling and max-pooling using a 7×7 filter size gives the spatial attention weight map $M_s \in R^{(1 \times h \times w)}$ as shown in equation 8:

$$\begin{aligned} \boldsymbol{F_s} &= \frac{1}{c}\textstyle\sum_{i \in c} \boldsymbol{F_c}(i) + \overset{max}{\scriptstyle i \in c}\boldsymbol{F_c}(i) \\ \boldsymbol{M_S} &= \sigma\big(f^{7 \times 7}(\boldsymbol{F_s})\big), \end{aligned} \tag{8}$$

where σ is the sigmoid activation function, $f^{7 \times 7}$ is a convolution with 7×7 filter size.

However, to lessen the number of parameters, CBAM uses dimensionality reduction to help manage the model's complexity. Nonlinear cross-channel relationships are captured throughout the dimensionality reduction process. The dimensionality reduction can lead to an inaccurate capture of interaction between channels. We adopted the ECA-Net approach [78] to solve this problem. ECA-Net uses global average pooling (GAP) to aggregate convolution features without reducing dimensionality. This is accomplished by increasing the number of parameters by a very modest amount while it successfully gathers details regarding cross-channel interactions and gains a substantial performance improvement. To understand channel attention, the ECA module estimates the kernel size K adaptively, then conducts a 1D convolution, and applies a Sigmoid function σ. The kernel size K can be adaptively determined as follows:

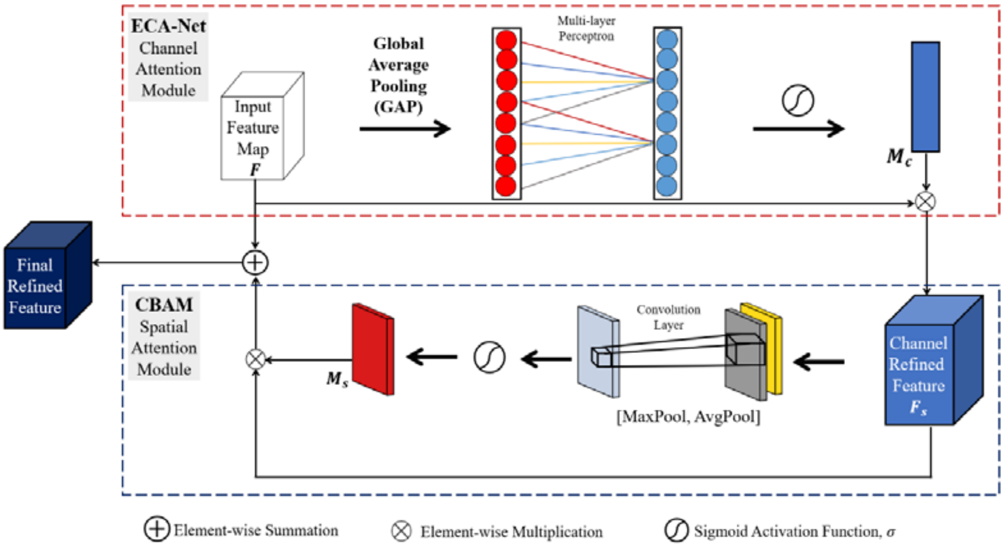

ECA-Net Channel Attention Module
Input Feature Map F
Global Average Pooling (GAP)
Multi-layer Perceptron
M_c
Final Refined Feature
CBAM Spatial Attention Module
M_s
Convolution Layer
[MaxPool, AvgPool]
Channel Refined Feature F_s
Element-wise Summation
Element-wise Multiplication
Sigmoid Activation Function, σ

Figure 6: The attention module architecture: The combination of ECA-Net and CBAM attention submodules to develop a complete channel and spatial attention mechanism.

$$K = \psi(C) = \left|\frac{log_2(c)}{\gamma} + \frac{b}{\gamma}\right|_{odd} \tag{9}$$

where $|t|_{odd}$ represents the nearest odd number of t, the kernel size K can be determined using mapping ψ and the number of channels (channel dimension) denoted by C. γ is set to 2, and b is set to 1.

In this work, we combined the ECA-Net and CBAM to achieve a powerful attention mechanism illustrated in Figure 6. We incorporated the combined ECA-Net/CBAM attention mechanism into the fusion layers of our proposed camera-radar fusion network (CR-YOLOnet) in Figure 7. The attention mechanism helps to draw more emphasis and improve the feature representation of the features that helps with object detection. We enhanced CR-YOLOnet with an attention framework to detect multi-scale object sizes in foggy weather conditions. The ECA-Net handled the channel submodule operations, while CBAM handled the spatial submodule operations. The ECA-Net module trains effectively on the input feature maps following a 1D convolutional GAP which generates the updated weight.

The channel-refined feature maps are produced from the element-wise multiplication of input feature maps and the updated weight. The output of the ECA module is sent into the CBAM's spatial attention module, which generates a 2D spatial attention map. The element-wise summation of the original input feature map and 2D spatial attention map is performed to obtain a residual-like architecture. The ReLU activation function is applied to the aggregated feature map to generate the final feature map sent to the detection head layer in Figure 7.

3.5. Proposed Camera-Radar Fusion Network (CR-YOLOnet)

We present our proposed network called CR-YOLOnet in Figure 7, a deep learning multiple sensor fusion object detector based on the baseline YOLOv5 network. To develop CR-YOLOnet, we made several adjustments to the baseline YOLOv5 model. Our CR-YOLOnet can take input from camera and radar sources compared to the single modal system in the baseline YOLOv5. There are two CSPDarknet backbone networks with which CR-YOLOnet extracts feature maps, each for the camera and radar sensors.

The feature information from the backbone network is sent to the feature fusion layers through two connections illustrated as round dot lines. The concepts of residual networks inspired the connections to improve the backpropagation of gradient in our network, prevent gradient fading, and minimize feature information loss for relatively tiny objects in fog.

As previously mentioned in section 3.4, we included the combined ECA-Net/CBAM attention mechanism into the fusion layers of CR-YOLOnet. The purpose of the attention mechanism is to enhance the capacity of CR-YOLOnet to detect multi-scale object sizes in medium and heavy foggy weather conditions. Especially small objects that are not nearby.

The detection head is made up of convolution blocks and utilizes all three scales of feature maps in the neck layer. The 2-dimensional convolution allows the detection head to produce three unique sets of detections, each having a resolution level of $80 \times 80 \times 12$, $80 \times 80 \times 12$, and $20 \times 20 \times 12$, respectively. The depth is 12 because the number of object classes is 7, the confidence level is 1, and 4 positional parameters. The total sum of which is 12.

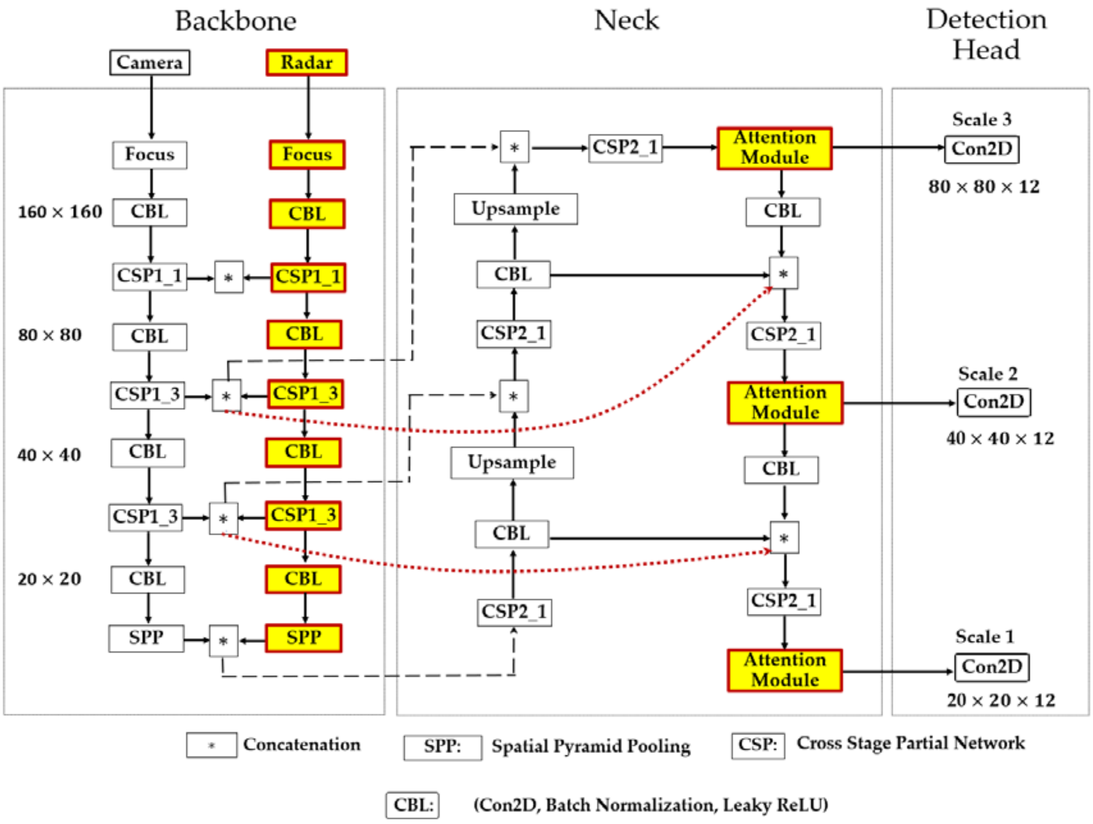


Figure 7: The architecture of our proposed CR-YOLOnet with attention module incorporated.

3.6 Loss Function

The three components comprise the loss function: (i) bounding box (position) loss, (ii) confidence loss, and (iii) classification loss. The bounding box loss function can be calculated when the intersection of the prediction box and the actual box is larger than the set threshold. The confidence loss and classification loss calculations are made when the object center enters the grid.

3.6.1. Bounding Box Loss Functions

We employed the complete intersection of union (CIoU) loss for bounding box regression [80]. Because the CIoU combines the following: (i) the overlap region between the predicted bounding box and the ground truth bounding box, (ii) the central point distance between the predicted bounding box and the ground truth bounding box, and (iii) the aspect ratio of the predicted bounding box and the ground truth bounding box. The CIoU approach combines these three components to improve the accuracy of average precision (AP) and average recall (AR) for object detection while achieving a faster convergence.

The CIoU loss function in equation 10 builds on the distance intersection of union (DIoU) loss [80] by enforcing a penalty term R_{CIoU} for the box aspect ratio given in equation 11.

$$L_{CIoU} = 1 - IoU + R_{CIoU} \tag{10}$$

$$R_{CIoU} = \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{11}$$

$$\alpha = \frac{v}{1 - IoU + v} \tag{12}$$

$$v = \frac{4}{\pi^2}\left(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}\right)^2 \tag{13}$$

where α is the weight function, a trade-off parameter that gives the overlap region factor a higher priority for regression, especially for non-overlapping cases, v helps to measure the consistency or similarity of the aspect ratio between the bounding boxes, b and b^{gt} are the central points of the predicted bounding box B and the ground-truth bounding box B^{gt}, the width and height of the predicted bounding and the ground-truth bounding boxes are denoted as w and h, and w^{gt} and h^{gt}, respectively.

3.6.2. Confidence Loss and Classification Loss Functions

The confidence loss function L_{obj} is as follows:

$$\begin{aligned} L_{obj} = & \sum_{i=0}^{s\times s}\sum_{j=0}^{b} I_{ij}^{obj}[\hat{C}_i \log(C_i) + (1 - \hat{C}_i)\,(1 - \log(C_i))] \\ & - \lambda_{noobj}\sum_{i=0}^{s\times s}\sum_{j=0}^{b} I_{ij}^{noobj}[\hat{C}_i \log(C_i) + (1 - \hat{C}_i)\,(1 - \log(C_i))] \end{aligned} \tag{14}$$

The classification loss function L_{cls} is as follows:

$$L_{cls} = \sum_{i=0}^{s\times s}\sum_{j=0}^{b} I_{ij}^{obj} \sum_{c \in classes} \left[\hat{P}_i(c)\log p_i(c) + (1 + \hat{P}_i(c))\log(1 - p_i(c))\right] \tag{15}$$

where I_{ij}^{obj} represents the object detected by the j^{th} boundary of grid cell, $s \times s$ denotes the number of grid points, b denotes the number of anchors associated with every grid, c denotes the number of categories, p represents the probability of categories, C denotes the box confidence score in cell i, $\widehat{C_\imath}$ denotes the box confidence score for the predicted object, λ_{noobj} denotes the weight representing the predicted loss of confidence in the bounding box in the absence of an object.

Therefore, the overall loss function is given as follows:

$$Loss = \sum_{i=0}^{s\times s} L_{CIoU} + L_{obj} + L_{cls} \tag{16}$$

4. Experimental Results

4.1 Dataset

In this work, we used the CARLA [81] simulator to create a simulated environment for autonomous driving from which we collected camera and radar data. Seven (7) different types of common road participants were included in our datasets. Since the camera observations and radar detections were associated, the radar detections were sparsely overlapped in white dots on the camera image, as shown in Figure 8.

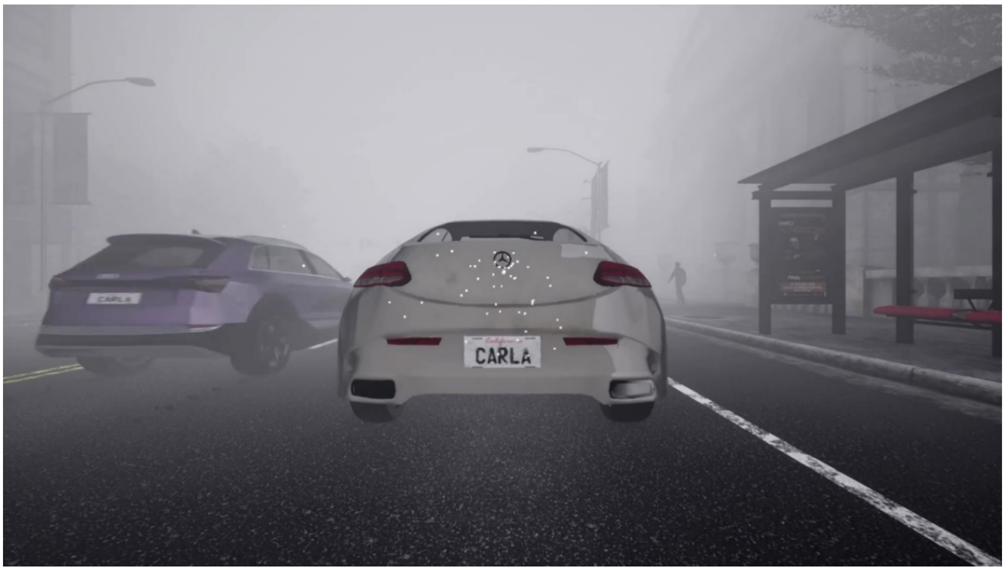


Figure 8: Camera and radar data obtain from CARLA simulator with radar data overlayed in white dot.

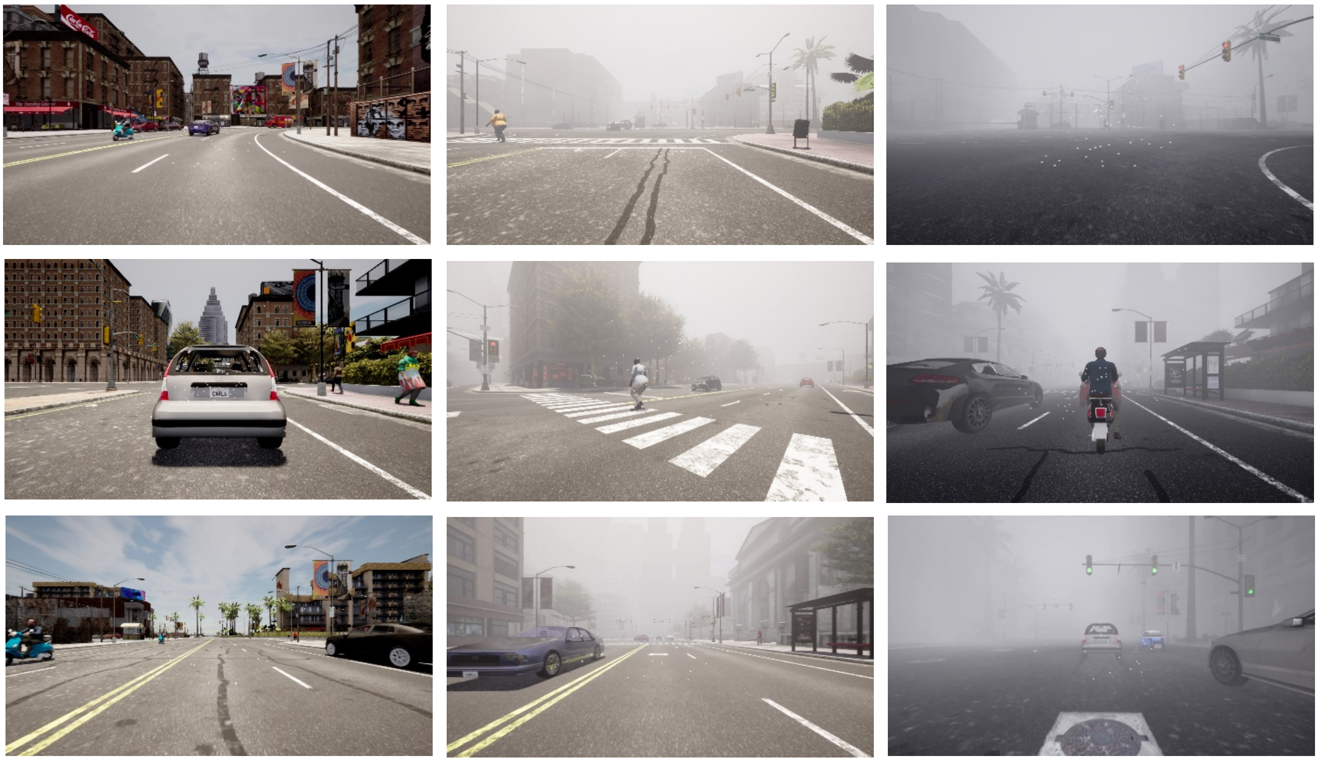


Figure 9: A Sample of our CARLA dataset showing clear day in the far-left column and varying level of fog in both columns to the right.

Figure 9 illustrates a sample of our CARLA dataset showing clear-day and varying fog levels. The visibility distance of fog ranges from 50m to 300m. The total number of images is 25000, with 80% (20000) belonging to the training set and with the remaining 5000 for testing and verification. We use clear and foggy weather conditions for CR-YOLOnet and YOLOv5 training and testing evaluations. Figure 10 shows the distribution of various object classes, including bicycle, bus, car, motorcycle, person, traffic light, and truck.

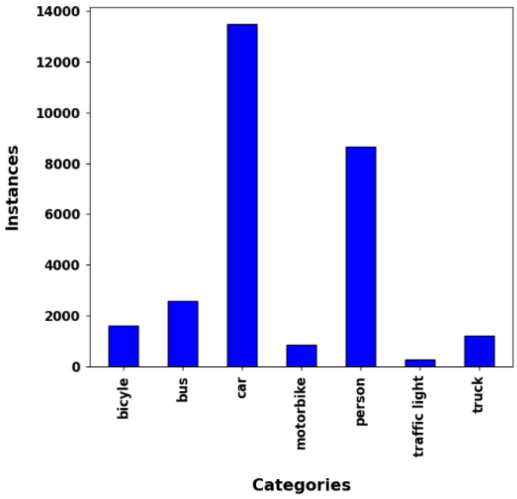


Figure 10: The distribution of various objects classes.

Table 1:Training parameters with clear only, fog only and clear + fog training sets.

Model	Model Size	Optimizer	Learning Rate	Weight Decay	Batch Size	Momentum	Epoch
YOLOv5	Small	Adam	0.0001	0.00025	64	0.821	300
YOLOv5	Medium	Adam	0.0001	0.00025	64	0.821	300
YOLOv5	Large	Adam	0.0001	0.00025	64	0.821	300
CR-YOLOnet	Small	Adam	0.0001	0.00025	64	0.821	300
CR-YOLOnet	Medium	Adam	0.0001	0.00025	64	0.821	300
CR-YOLOnet	Large	Adam	0.0001	0.00025	64	0.821	300

4.2 Experimental platform and training parameters

The PyTorch framework was used to conduct the experiment in Python programming. The hardware and software settings are as follows: Graphics card: Nvidia GeForce RTX 2070 with Max-Q Design; RAM: 16 gigabytes of memory; CPU: Intel Core 17-8570H 2.2 GHz 6 cores. Table 1 illustrates the parameters for the three different model sizes (small, medium, and large) CR-YOLOnet and baseline YOLOv5 models were trained. With only about 7.5 million parameters, YOLOv5s is a small but fast model, making it well-suited for inference on the central processing unit.

The YOLOv5m model is considered medium-sized with its 21.5 million parameters because it strikes an outstanding balance between speed and accuracy. Among the YOLOv5 derivatives, YOLOv5l is the largest, with a total of 46.8 million parameters. It is efficient for the detection of small objects. The CR-YOLOnet was trained on both the image and radar data and only image data for YOLOv5. To begin with, the rate of learning steadily increases, and then it gradually decreases. The network's utilization of the pre-training rate causes the increased learning process at the beginning. Each model was trained using clear only, fog only, and clear + fog datasets for 300 epochs with a batch size of 64, weight decay of 0.00025, a learning rate of 0.0001, and a learning rate momentum of 0.821 using Adam optimization.

4.3 Evaluation Metrics

Deep learning can be evaluated using a variety of metrics, including accuracy, confusion matrix, precision, recall, average precision, mean average precision (map), intersection union ratio, and average precision. In this work, we use the same set of evaluation metrics as the COCO dataset [66], precision, recall, and average precision (AP) for small (APs), medium (APm), and large (APl) object areas. Also, we estimated F1 score and mean average precision (mAP) thresholds at 0.5. We compared the performance of our CR-YOLOnet to the baseline YOLOv5 in clear, light, medium, and foggy environments for small, medium, and large model sizes. Equations 17 to 21 describe the evaluation metrics.

Precision P can be expressed as:

$$P = \frac{TP}{TP + FP} \tag{17}$$

Recall R can be expressed as:

$$R = \frac{TP}{TP + FN} \tag{18}$$

The F1 score can be expressed as:

$$F1 = \frac{2(P \times R)}{P + R} \tag{19}$$

where TP denotes the outcome that occurs when the category of an object is accurately identified in an image, FP represents the outcome that occurs when the category of an object is inaccurately identified in an image, FN is the outcome that occurs when an attempt to identify an object in an image failed.

The average precision (AP) is the area under the $Precision - Recall$ curve with values between 0 and 1, and it is expressed in equation 17:

$$AP = \int_0^1 \mathrm{P(R)dR} \tag{20}$$

The mean average precision (mAP) is the mean for all N categories evaluated in the dataset, and it can be estimated as follows in equation 18:

$$mAP = \frac{1}{\mathrm{N}} \sum_{i=1}^{N} AP_i \tag{21}$$

4.4 Training Results and Discussion

To ensure the algorithm's detection efficiency, our improved method (CR-YOLOnet) is compared to the baseline YOLOv5. A contrast of the changes in mAP that occur throughout the training process for our CR-YOLOnet and the YOLOv5 (small, medium, large) models can be seen in Figure 11. Each CR-YOLOnet model was trained on radar and image (clear only, fog only and fog + clear) training sets. When compared to the YOLOv5 network, the rise in mAP experienced by the CR-YOLO network was stable and much quicker due to the multi-sensor integration advantage.

The CR-YOLOnet large model, as shown in Table 2, clearly achieves the highest performance, with a $F1$ of 0.861, recall of 0.885, the precision of 0.914, and mAP of 0.896. However, the network that strikes a balance between accuracy and speed best is our CR-YOLOnet small model with an mAP of 0.849 and 69 frames per second.

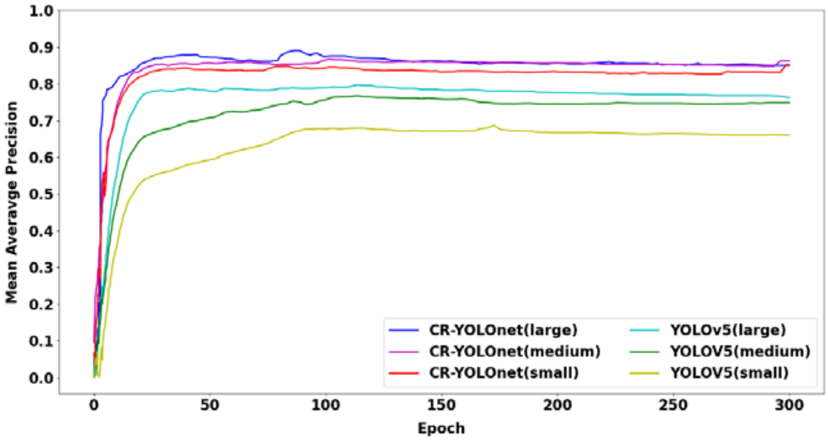


Figure 11: Comparison of mAP (0.5) for CR-YOLOnet and YOLOv5.

Table 2: Performance Comparison of our CR-YOLOnet and YOLOv5.

Model	Model Size	F1	Recall	Precision	mAP(0.5)	mAP Contrast	FPS
YOLOv5	Small	0.714	0.719	0.705	0.685	baseline	98
YOLOv5	Medium	0.692	0.776	0.738	0.771	↑0.086	46
YOLOv5	Large	0.756	0.813	0.792	0.795	↑0.110	25
CR-YOLOnet	Small	0.821	0.805	0.839	**0.849**	**↑0.164**	**69**
CR-YOLOnet	Medium	0.829	0.830	0.844	0.862	↑0.177	52
CR-YOLOnet	Large	0.861	0.885	0.914	**0.896**	↑0.211	27

4.4 Testing Results and Discussion

Observing the model's performance in various clear-day and foggy weather conditions is essential for establishing its reliability. Tables 3 to 5 show the comparison of detection AP for small, medium, and large object areas and mAP at $IoU = 0.5$. The comparison was made for large (Table 3), medium (Table 4), and small (Table 5) model sizes under clear, light, medium, and heavy fog conditions.

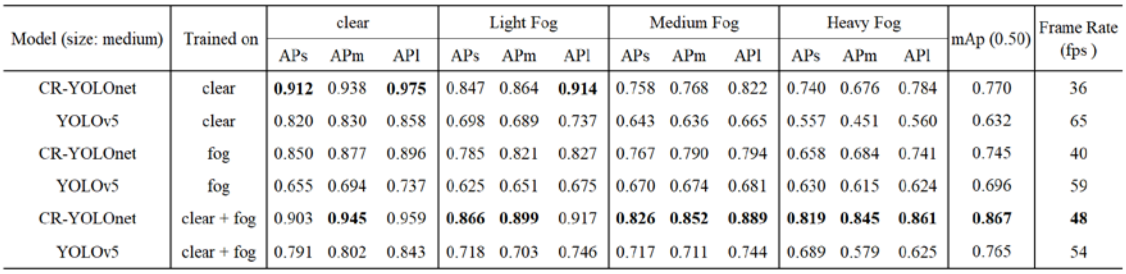
In Table 3, CR-YOLOnet trained on clear-day datasets performed best under clear weather conditions with APs of 0.928 and APl of 0.989. However, CR-YOLOnet trained on clear + fog datasets performed better than the other five models, with the highest mAP of 0.892 for clear and foggy conditions. An improvement of 11.78% in mAP and when compared to YOLOv5 trained on clear + fog with mAP of 0.798. Table 4 shows that CR-YOLOnet with APs of 0.912 and APl of 0.975 performs best in clear weather when it was trained on clear-day datasets. Out of the six models tested, CR-YOLOnet trained on clear + fog datasets had the most significant (mAP) of 0.867. An improvement of 13.33% in mAP when compared to YOLOv5 trained on clear + fog with mAP of 0.765.

Table 3: Comparison of detection AP for small, medium, and large object areas and mAP(0.5) using large model size

Model (size: large)	Trained on	clear			Light Fog			Medium Fog			Heavy Fog			mAp (0.50)	Frame Rate (fps)
		APs	APm	APl	APs	APm	APl	APs	APm	APl	APs	APm	APl		
CR-YOLOnet	clear	**0.928**	0.957	**0.989**	0.903	0.928	**0.936**	0.808	0.817	0.871	0.791	0.727	0.833	0.815	22
YOLOv5	clear	0.833	0.856	0.877	0.698	0.727	0.783	0.679	0.693	0.728	0.611	0.505	0.613	0.745	28
CR-YOLOnet	fog	0.845	0.864	0.885	0.816	0.863	0.872	0.802	0.815	0.833	0.709	0.735	0.792	0.766	19
YOLOv5	fog	0.625	0.721	0.741	0.594	0.650	0.716	0.711	0.725	0.743	0.682	0.667	0.676	0.717	25
CR-YOLOnet	clear + fog	0.921	**0.965**	0.972	**0.912**	**0.923**	0.949	**0.833**	**0.884**	**0.920**	**0.851**	**0.877**	**0.893**	**0.892**	**23**
YOLOv5	clear + fog	0.795	0.869	0.883	0.717	0.739	0.755	0.748	0.769	0.806	0.740	0.632	0.677	0.798	25

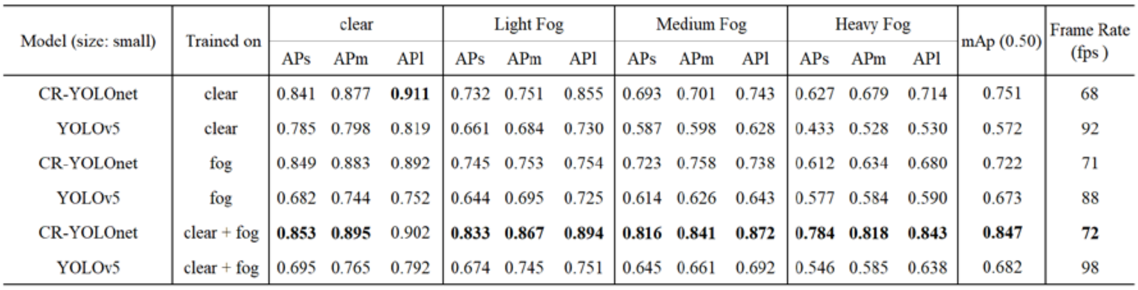
doi:10.20944/preprints202305.2180.v1

Table 4: Comparison of detection AP for small, medium, and large object areas and mAP(0.5) using medium model size.



Model (size: medium)	Trained on	clear			Light Fog			Medium Fog			Heavy Fog			mAp (0.50)	Frame Rate (fps)
		APs	APm	APl	APs	APm	APl	APs	APm	APl	APs	APm	APl		
CR-YOLOnet	clear	**0.912**	0.938	**0.975**	0.847	0.864	**0.914**	0.758	0.768	0.822	0.740	0.676	0.784	0.770	36
YOLOv5	clear	0.820	0.830	0.858	0.698	0.689	0.737	0.643	0.636	0.665	0.557	0.451	0.560	0.632	65
CR-YOLOnet	fog	0.850	0.877	0.896	0.785	0.821	0.827	0.767	0.790	0.794	0.658	0.684	0.741	0.745	40
YOLOv5	fog	0.655	0.694	0.737	0.625	0.651	0.675	0.670	0.674	0.681	0.630	0.615	0.624	0.696	59
CR-YOLOnet	clear + fog	0.903	**0.945**	0.959	**0.866**	**0.899**	0.917	**0.826**	**0.852**	**0.889**	**0.819**	**0.845**	**0.861**	**0.867**	**48**
YOLOv5	clear + fog	0.791	0.802	0.843	0.718	0.703	0.746	0.717	0.711	0.744	0.689	0.579	0.625	0.765	54

Table 5: Comparison of detection AP for small, medium, and large object areas and mAP(0.5) using small model size.



Model (size: small)	Trained on	clear			Light Fog			Medium Fog			Heavy Fog			mAp (0.50)	Frame Rate (fps)
		APs	APm	APl	APs	APm	APl	APs	APm	APl	APs	APm	APl		
CR-YOLOnet	clear	0.841	0.877	**0.911**	0.732	0.751	0.855	0.693	0.701	0.743	0.627	0.679	0.714	0.751	68
YOLOv5	clear	0.785	0.798	0.819	0.661	0.684	0.730	0.587	0.598	0.628	0.433	0.528	0.530	0.572	92
CR-YOLOnet	fog	0.849	0.883	0.892	0.745	0.753	0.754	0.723	0.758	0.738	0.612	0.634	0.680	0.722	71
YOLOv5	fog	0.682	0.744	0.752	0.644	0.695	0.725	0.614	0.626	0.643	0.577	0.584	0.590	0.673	88
CR-YOLOnet	clear + fog	**0.853**	**0.895**	0.902	**0.833**	**0.867**	**0.894**	**0.816**	**0.841**	**0.872**	**0.784**	**0.818**	**0.843**	**0.847**	**72**
YOLOv5	clear + fog	0.695	0.765	0.792	0.674	0.745	0.751	0.645	0.661	0.692	0.546	0.585	0.638	0.682	98

Table 6: Comparison of detection AP per object class.

Model Trained on: clear + fog	Model Size	Bicycle	Bus	Car	Motorcycle	Pedestrians	Traffic Light	Truck
CR-YOLOnet	Small	0.711	0.742	0.747	0.729	0.736	0.751	0.725
YOLOv5	Small	0.595	0.606	0.634	0.596	0.609	0.632	0.597
CR-YOLOnet	Medium	0.779	0.835	0.821	0.809	0.827	0.818	0.815
YOLOv5	Medium	0.683	0.692	0.715	0.684	0.695	0.713	0.685
CR-YOLOnet	Large	**0.886**	**0.857**	**0.890**	**0.885**	**0.901**	**0.863**	**0.847**
YOLOv5	Large	0.761	0.745	0.763	0.695	0.790	0.748	0.756

In Table 5, CR-YOLOnet trained on clear + fog datasets outperformed the other five models in almost all metrics. In Table 6, we illustrate the comparison of detection AP per object class. The CR-YOLOnet trained on clear + fog datasets outperformed the other five models for each object class. However, compared to the large (Table 3) and medium (Table 4) models, the CR-YOLOnet trained on clear + fog datasets in Table 5 struck a balance between accuracy and speed with mAP of 0.847 and speed of 72 FPS for both clear and foggy circumstances. This implies that our CR-YOLOnet small model trained on clear + fog datasets has the best capacity to accurately detect small objects in fog without trading off speed.

As a result, in Figure 12, we compare the qualitative results of our small CR-YOLOnet and mediumYOLOv5 models, with both models trained on clear + fog datasets. We selected the medium YOLOv5 model trained on clear + fog datasets because it struck a balance between speed and accuracy, as illustrated in Table 4. Figure 12 (a) shows the input data with varying visibility and proximity of close objects at about 50m and most distant objects at 300m. Figure 12 (b) and 12 (c) show the detection results of medium YOLOv5 and small CR-YOLOnet models, respectively. Both models could detect objects in close proximity. However, only our small model CR-YOLOnet trained on clear + fog datasets could detect objects beyond 100m in medium and 75m heavy foggy conditions.

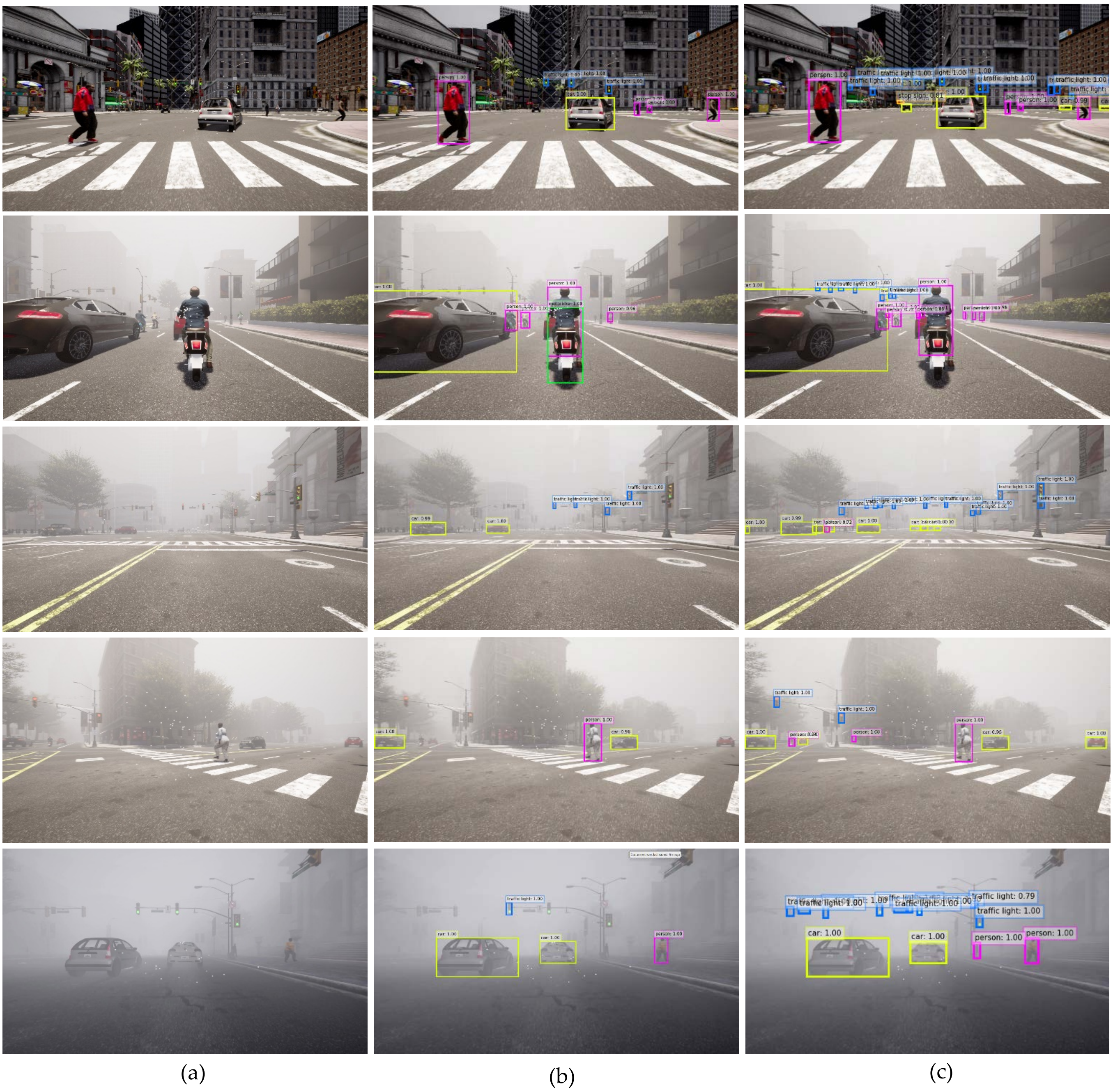


Figure 12: Comparison of the qualitative results of our CR-YOLOnet and baseline YOLOv5 (a) input data with varying visibility and proximity: from clear (top) to heavy fog (bottom (b) results of Medium YOLOv5 model trained on clear + fog, (c) results of small CR-YOLOnet model trained on clear + fog.

5. Conclusion

In this paper, we introduced an enhanced YOLOv5-based multi-sensor fusion network (CR-YOLOnet) that fused radar object identification with camera image bounding box to locate and identify small and distant objects in fog. We transformed the radar detections by mapping them into two-dimensional image coordinates and projected the resulting radar image onto the camera image. Using image data, we demonstrated that atmospheric distortion has a negative impact on sensor data in fog. We showed that our CR-YOLOnet. In contrast to the single-modal system used in the baseline YOLOv5, we showed that our CR-YOLOnet is capable of receiving data from both the camera and radar sources. The CR-YOLOnet utilized two different CSPDarknet backbone networks feature maps feature extraction, one for the camera sensors and another for the radar sensors.

We emphasized and improved critical feature representation required for object detection using attention mechanisms and introduced two residual-like connections to reduce high-level feature information loss. We simulated autonomous driving instances under clear and foggy weather conditions in the CARLA simulator to obtain clear and multi-fog weather datasets. We implemented our CR-YOLOnet and the baseline YOLOv5 in model configurations of three sizes (small, medium, and large). We found that both small CR-YOLOnet and medium YOLOv5 trained on clear + fog datasets struck a balance between speed and accuracy with a mAP of 0.847 and a speed of 72 FPS. There was an improvement of 24.19% in mAP compared to YOLOv5 trained on clear + fog with a mAP of 0.765. However, the performance of CR-YOLOnet was significantly improved, especially in medium and heavy fog conditions. Since the large YOLOv5 model is more efficient for the detection of small objects, in the future, we can optimize the speed of our large CR-YOLOnet by reducing the dimension of the input data, using half-precision floating points that lower the memory usage in neural networks, enhance the backbone network with attention mechanism, etc. without trading off its accuracy.

Author Contributions: Conceptualization, I.O. and S.B.; methodology, I.O.; software, I.O.; writing—original draft preparation, I.O.; writing—review and editing, I.O.; supervision and review, S.B. All authors have read and agreed to the published version of the manuscript.

Funding: Please add: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. Ogunrinde, I.; Bernadin, S. A Review of the Impacts of Defogging on Deep Learning-Based Object Detectors in Self-Driving Cars. In Proceedings of the SoutheastCon 2021, 10-13 March 2021, 2021; pp. 01-08.
2. Liu, Z.; He, Y.; Wang, C.; Song, R.J.S. Analysis of the influence of foggy weather environment on the detection effect of machine vision obstacles. **2020**, *20*, 349.
3. Zang, S.; Ding, M.; Smith, D.; Tyler, P.; Rakotoarivelo, T.; Kaafar, M.A. The Impact of Adverse Weather Conditions on Autonomous Vehicles: How Rain, Snow, Fog, and Hail Affect the Performance of a Self-Driving Car. *IEEE Vehicular Technology Magazine* **2019**, *14*, 103-111.
4. Hamzeh, Y.; El-Shair, Z.; Rawashdeh, S.A. Effect of Adherent Rain on Vision-Based Object Detection Algorithms. 2020.
5. Wu, J.; Xu, H.; Tian, Y.; Pi, R.; Yue, R.J.S. Vehicle detection under adverse weather from roadside LiDAR data. **2020**, *20*, 3433.
6. Kim, T.-L.; Park, T.-H. Camera-LiDAR Fusion Method with Feature Switch Layer for Object Detection Networks. *Sensors* **2022**, *22*, 7163.
7. Miclea, R.-C.; Ungureanu, V.-I.; Sandru, F.-D.; Silea, I. Visibility Enhancement and Fog Detection: Solutions Presented in Recent Scientific Papers with Potential for Application to Mobile Systems. *Sensors* **2021**, *21*, 3370.
8. Lee, J.; Shiotsuka, D.; Nishimori, T.; Nakao, K.; Kamijo, S. GAN-Based LiDAR Translation between Sunny and Adverse Weather for Autonomous Driving and Driving Simulation. *Sensors* **2022**, *22*, 5287.
9. Younis, R.; Bastaki, N. Accelerated Fog Removal from Real Images for Car Detection. In Proceedings of the 2017 9th IEEE-GCC Conference and Exhibition (GCCCE), 2017; pp. 1-6.

10. *Federal Meteorological Handbook Number 1: Chapter 8-Present Weather*; Office of the Federal Coordinator for Meteorology: 2005; Volume vol. 8.
11. Abdu, F.J.; Zhang, Y.; Fu, M.; Li, Y.; Deng, Z. Application of Deep Learning on Millimeter-Wave Radar Signals: A Review. **2021**, *21*, 1951.
12. De Ponte Müller, F. Survey on Ranging Sensors and Cooperative Techniques for Relative Positioning of Vehicles. **2017**, *17*, 271.
13. Choi, W.Y.; Yang, J.H.; Chung, C.C. Data-Driven Object Vehicle Estimation by Radar Accuracy Modeling with Weighted Interpolation. **2021**, *21*, 2317.
14. Nabati, R.; Qi, H.J.A. Radar-Camera Sensor Fusion for Joint Object Detection and Distance Estimation in Autonomous Vehicles. **2020**, *abs/2009.08428*.
15. Magosi, Z.F.; Li, H.; Rosenberger, P.; Wan, L.; Eichberger, A. A Survey on Modelling of Automotive Radar Sensors for Virtual Test and Validation of Automated Driving. *Sensors* **2022**, *22*, 5693.
16. Joshi, K.A.; Thakore, D.G.J.I.J.o.S.C.; Engineering. A survey on moving object detection and tracking in video surveillance system. **2012**, *2*, 44-48.
17. Cooney, M.; Bigun, J. PastVision: Exploring" Seeing" into the Near Past with a Thermal Camera and Object Detection--For Robot Monitoring of Medicine Intake by Dementia Patients. **2017**.
18. Lin, M.C. Efficient collision detection for animation and robotics. PhD thesis, Department of Electrical Engineering and Computer Science …, 1993.
19. Li, Z.; Dong, M.; Wen, S.; Hu, X.; Zhou, P.; Zeng, Z.J.N. CLU-CNNs: Object detection for medical images. **2019**, *350*, 53-59.
20. Bhat, S.; Meenakshi, M. Vision Based Robotic System for Military Applications--Design and Real Time Validation. In Proceedings of the 2014 Fifth International Conference on Signal and Image Processing, 2014; pp. 20-25.
21. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the CACM, 2017.
22. Jocher, G.; Nishimura, K.; Mineeva, T.; Vilariño, R. YOLOv5 (2020). *GitHub repository: https://github. com/ultralytics/yolov5* **2020**.
23. Zhou, Y.; Liu, L.; Zhao, H.; López-Benítez, M.; Yu, L.; Yue, Y. Towards Deep Radar Perception for Autonomous Driving: Datasets, Methods, and Challenges. *Sensors* **2022**, *22*, 4208.
24. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A Multimodal Dataset for Autonomous Driving. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* **2020**, 11618-11628.
25. Barnes, D.; Gadd, M.; Murcutt, P.; Newman, P.; Posner, I. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020; pp. 6433-6438.
26. Kim, G.; Park, Y.S.; Cho, Y.; Jeong, J.; Kim, A. MulRan: Multimodal Range Dataset for Urban Place Recognition. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), 31 May-31 Aug. 2020, 2020; pp. 6246-6253.
27. Sheeny, M.; De Pellegrin, E.; Mukherjee, S.; Ahrabian, A.; Wang, S.; Wallace, A. RADIATE: A radar dataset for automotive perception in bad weather. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021; pp. 1-7.
28. Meyer, M.; Kuschk, G. Automotive radar dataset for deep learning based 3d object detection. In Proceedings of the 2019 16th european radar conference (EuRAD), 2019; pp. 129-132.

29. Bijelic, M.; Gruber, T.; Mannan, F.; Kraus, F.; Ritter, W.; Dietmayer, K.; Heide, F. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020; pp. 11682-11692.

30. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* **2020**.

31. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V.J.a.p.a. CARLA: An open urban driving simulator. **2017**.

32. Nabati, M.R. Sensor Fusion for Object Detection and Tracking in Autonomous Vehicles. Dissertation, University of Tennessee, Knoxville, Knoxville, 2021.

33. Ahmed, M.; Hashmi, K.A.; Pagani, A.; Liwicki, M.; Stricker, D.; Afzal, M.Z. Survey and Performance Analysis of Deep Learning Based Object Detection in Challenging Environments. **2021**, *21*, 5116.

34. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2014; pp. 580-587.

35. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), 7-13 Dec. 2015, 2015; pp. 1440-1448.

36. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2017**, *39*, 1137-1149.

37. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2016; pp. 779-788.

38. Redmon, J.; Farhadi, A. YOLO9000: better, faster, stronger. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2017; pp. 7263-7271.

39. Redmon, J.; Farhadi, A.J.a.p.a. Yolov3: An incremental improvement. **2018**.

40. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2017; pp. 2117-2125.

41. 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 European conference on computer vision, 2016; pp. 21-37.

42. Walambe, R.; Marathe, A.; Kotecha, K.; Ghinea, G. Lightweight Object Detection Ensemble Framework for Autonomous Vehicles in Challenging Weather Conditions. *Computational Intelligence and Neuroscience* **2021**, *2021*, 5278820, doi:10.1155/2021/5278820.

43. Gruber, T.; Bijelic, M.; Ritter, W.; Dietmayer, K.C.J. Gated Imaging for Autonomous Driving in Adverse Weather. 2019.

44. Tumas, P.; Nowosielski, A.; Serackis, A. Pedestrian Detection in Severe Weather Conditions. *IEEE Access* **2020**, *8*, 62775-62784.

45. Sommer, L.; Acatay, O.; Schumann, A.; Beyerer, J. Ensemble of Two-Stage Regression Based Detectors for Accurate Vehicle Detection in Traffic Surveillance Data. In Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 27-30 Nov. 2018, 2018; pp. 1-6.

46. Sindagi, V.A.; Oza, P.; Yasarla, R.; Patel, V.M. Prior-Based Domain Adaptive Object Detection for Hazy and Rainy Conditions. In Proceedings of the Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV, Glasgow, United Kingdom, 2020; pp. 763–780.

47. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2018; pp. 8759-8768.

48. Lombacher, J.; Hahn, M.; Dickmann, J.; Wöhler, C. Potential of radar for static object classification using deep learning methods. In Proceedings of the 2016 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM), 19-20 May 2016, 2016; pp. 1-4.

49. Palffy, A.; Dong, J.; Kooij, J.F.P.; Gavrila, D.M. CNN Based Road User Detection Using the 3D Radar Cube. *IEEE Robotics and Automation Letters* **2020**, *5*, 1263-1270, doi:10.1109/LRA.2020.2967272.

50. Lee, S. Deep learning on radar centric 3d object detection. *arXiv preprint arXiv:2003.00851* **2020**.

51. Nabati, R.; Qi, H. Rrpn: Radar region proposal network for object detection in autonomous vehicles. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), 2019; pp. 3093-3097.

52. Chadwick, S.; Maddern, W.; Newman, P. Distant vehicle detection using radar and vision. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), 2019; pp. 8311-8317.

53. Nobis, F.; Geisslinger, M.; Weber, M.; Betz, J.; Lienkamp, M. A deep learning-based radar and camera sensor fusion architecture for object detection. In Proceedings of the 2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF), 2019; pp. 1-7.

54. Meyer, M.; Kuschk, G. Deep learning based 3d object detection for automotive radar and camera. In Proceedings of the 2019 16th European Radar Conference (EuRAD), 2019; pp. 133-136.

55. Zhang, X.; Zhou, M.; Qiu, P.; Huang, Y.; Li, J. Radar and vision fusion for the real-time obstacle detection and identification. *Industrial Robot: the international journal of robotics research and application* **2019**.

56. John, V.; Mita, S. RVNet: Deep sensor fusion of monocular camera and radar for image-based obstacle detection in challenging environments. In Proceedings of the Pacific-Rim Symposium on Image and Video Technology, 2019; pp. 351-364.

57. John, V.; Nithilan, M.; Mita, S.; Tehrani, H.; Sudheesh, R.; Lalu, P. So-net: Joint semantic segmentation and obstacle detection using deep fusion of monocular camera and radar. In Proceedings of the Pacific-Rim Symposium on Image and Video Technology, 2019; pp. 138-148.

58. Zhou, T.; Jiang, K.; Xiao, Z.; Yu, C.; Yang, D. Object Detection Using Multi-Sensor Fusion Based on Deep Learning. In *CICTP 2019*; 2019; pp. 5770-5782.

59. Chang, S.; Zhang, Y.; Zhang, F.; Zhao, X.; Huang, S.; Feng, Z.; Wei, Z. Spatial Attention Fusion for Obstacle Detection Using MmWave Radar and Vision Sensor. *Sensors* **2020**, *20*, 956.

60. Bai, J.; Li, S.; Zhang, H.; Huang, L.; Wang, P. Robust Target Detection and Tracking Algorithm Based on Roadside Radar and Camera. *Sensors* **2021**, *21*, 1116.

61. Zhang, X.; Zhou, M.; Qiu, P.; Huang, Y.; Li, J. Radar and vision fusion for the real-time obstacle detection and identification. *Industrial Robot: the international journal of robotics research and application* **2019**, *46*, 391-395, doi:10.1108/IR-06-2018-0113.

62. Koschmieder, H.J.B.z.P.d.f.A. Theorie der horizontalen Sichtweite. **1924**, 33-53.

63. Narasimhan, S.G.; Nayar, S.K.J.I.j.o.c.v. Vision and the atmosphere. **2002**, *48*, 233-254.

64. HE, K. Single Image Haze Removal Using Dark Channel Prior. Dissertation, The Chinese University of Hong Kong, 2011.

65. Mai, N.A.M.; Duthon, P.; Khoudour, L.; Crouzil, A.; Velastín, S.A. 3D Object Detection with SLS-Fusion Network in Foggy Weather Conditions. *Sensors (Basel)* **2021**, *21*.

66. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European conference on computer vision, 2014; pp. 740-755.

67. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The PASCAL visual object classes challenge 2007 (VOC2007) results. **2007**.

68. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* **2018**.

69. 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 Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020; pp. 390-391.

70. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence* **2015**, *37*, 1904-1916.

71. Duvenaud, D.; Rippel, O.; Adams, R.; Ghahramani, Z. Avoiding pathologies in very deep networks. In Proceedings of the Artificial Intelligence and Statistics, 2014; pp. 202-210.

72. Saxe, A.M.; McClelland, J.L.; Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120* **2013**.

73. Fan, Y.; Liu, J.; Yao, R.; Yuan, X. COVID-19 Detection from X-ray Images using Multi-Kernel-Size Spatial-Channel Attention Network. *Pattern Recognition* **2021**, *119*, 108055, doi:https://doi.org/10.1016/j.patcog.2021.108055.

74. Li, G.; Fang, Q.; Zha, L.; Gao, X.; Zheng, N. HAM: Hybrid attention module in deep convolutional neural networks for image classification. *Pattern Recognition* **2022**, *129*, 108785, doi:https://doi.org/10.1016/j.patcog.2022.108785.

75. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual Attention Network for Image Classification. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 21-26 July 2017, 2017; pp. 6450-6458.

76. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the Computer Vision – ECCV 2018, Cham, 2018//, 2018; pp. 3-19.

77. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2020**, *42*, 2011-2023, doi:10.1109/TPAMI.2019.2913372.

78. Wang, Q.; Wu, B.; Zhu, P.F.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* **2020**, 11531-11539.

79. Zhu, L.; Geng, X.; Li, Z.; Liu, C. Improving YOLOv5 with Attention Mechanism for Detecting Boulders from Planetary Images. *Remote Sensing* **2021**, *13*, 3776.

80. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the Proceedings of the AAAI conference on artificial intelligence, 2020; pp. 12993-13000.

81. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An open urban driving simulator. In Proceedings of the Conference on robot learning, 2017; pp. 1-16.