



Article

Adaptive Feature Attention Module for Robust Visual–LiDAR Fusion-Based Object Detection in Adverse Weather Conditions

Taek-Lim Kim ^{1,†} , Saba Arshad ^{2,†} and Tae-Hyoung Park ^{3,*} ¹ Department of Control and Robot Engineering, Chungbuk National University, Cheongju 28644, Republic of Korea; taeglem@cbnu.ac.kr² Industrial Artificial Intelligence Research Center, Chungbuk National University, Cheongju 28644, Republic of Korea; sabarshad1000@gmail.com³ Department of Intelligent Systems & Robotics, Chungbuk National University, Cheongju 28644, Republic of Korea

* Correspondence: taehpark@cbnu.ac.kr

† These authors contributed equally to this work.

Abstract: Object detection is one of the vital components used for autonomous navigation in dynamic environments. Camera and lidar sensors have been widely used for efficient object detection by mobile robots. However, they suffer from adverse weather conditions in operating environments such as sun, fog, snow, and extreme illumination changes from day to night. The sensor fusion of camera and lidar data helps to enhance the overall performance of an object detection network. However, the diverse distribution of training data makes the efficient learning of the network a challenging task. To address this challenge, we systematically study the existing visual and lidar features based on object detection methods and propose an adaptive feature attention module (AFAM) for robust multisensory data fusion-based object detection in outdoor dynamic environments. Given the camera and lidar features extracted from the intermediate layers of EfficientNet backbones, the AFAM computes the uncertainty among the two modalities and adaptively refines visual and lidar features via attention along the channel and the spatial axis. The AFAM integrated with the EfficientDet performs the adaptive recalibration and fusion of visual lidar features by filtering noise and extracting discriminative features for an object detection network under specific environmental conditions. We evaluate the AFAM on a benchmark dataset exhibiting weather and light variations. The experimental results demonstrate that the AFAM significantly enhances the overall detection accuracy of an object detection network.

Keywords: multi-sensor fusion; deep fusion; object detection; deep learning

Citation: Kim, T.-L.; Arshad, S.; Park, T.-H. Adaptive Feature Attention Module for Robust Visual–LiDAR Fusion-Based Object Detection in Adverse Weather Conditions. *Remote Sens.* **2023**, *15*, 3992. <https://doi.org/10.3390/rs15163992>

Academic Editor: Pinliang Dong

Received: 21 June 2023

Revised: 1 August 2023

Accepted: 8 August 2023

Published: 11 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Autonomous navigation aims to enable safe driving without human intervention. It relies on various element technologies, including simultaneous localization and mapping (SLAM) [1], 3D pose estimation, object classification, detection [2], etc., to perceive and understand the surrounding environment. In particular, object detection plays a crucial role in autonomous navigation by detecting obstacles, pedestrians, and other vehicles on the road and making informed decisions to ensure safe driving [3].

To achieve reliable and consistent performance in object detection, even in a dynamic environment, researchers often propose sensor fusion, a technique that integrates data from multiple sensors. When noise increases in one sensor data, sensor fusion compensates for performance degradation by combining information from other sensors. However, developing a deep learning network for sensor fusion requires updating the network parameters to extract key features. When the sensor data significantly changes, such as when a different camera is used or the environmental conditions change, the feature extraction process may not be consistent, leading to performance degradation [4].

To address the aforementioned problem, effective sensor fusion must consider the impact of data changes on performance degradation. This involves developing robust feature extraction methods that are resilient to changes in sensor data and environmental conditions. By incorporating these methods, the sensor fusion approach can continue to deliver consistent and reliable performance, even in a dynamic environment, and thus enable safe and effective autonomous navigation.

Numerous methods have been proposed in the past focusing on the development of a multi-sensor fusion-based object detection system [4–10]. Such systems mainly consist of two major components i.e., robust feature extraction and data fusion. In a network-based multi-sensor system using deep fusion, each sensor dataset is processed by an appropriate network separately to extract features. For the extraction of robust features, network configurations are used that are optimized for each sensor, i.e., camera and lidar. This approach enables the extraction of robust features from sensor data. Later, those features are then fused in the middle of the network using various network structures. The convergence of extracted robust features leads to the improved performance of an object detection network.

Following the benefits of a deep fusion approach [9,11], we propose an adaptive feature attention module for an object detection network with which we extract the visual and lidar features from camera and lidar data using respective networks. The extraction of distinctive and robust features is performed using an attention mechanism in the middle layer of the fusion network. This is achieved by selecting the maximum or average value from the channel of the tensor, which reduces high-dimensional features to low-dimensional ones. Those features are then merged in the middle of the network, enabling the sensor data to be converged into a unified feature representation. During the feature space merging process, some features may cause noise during network training. Moreover, in the case of adverse weather conditions, such as fog, snow, sun, or illumination variations from day to night, sensory noise is inevitable. For example, lidar sensor noise increases in case of fog, while visual feature detection is difficult at night in comparison to daytime. In such scenarios, robust feature extraction is a challenging task [12]. To address this issue, the proposed AFAM performs the adaptive filtration of unnecessary features causing noise, as illustrated in Figure 1.

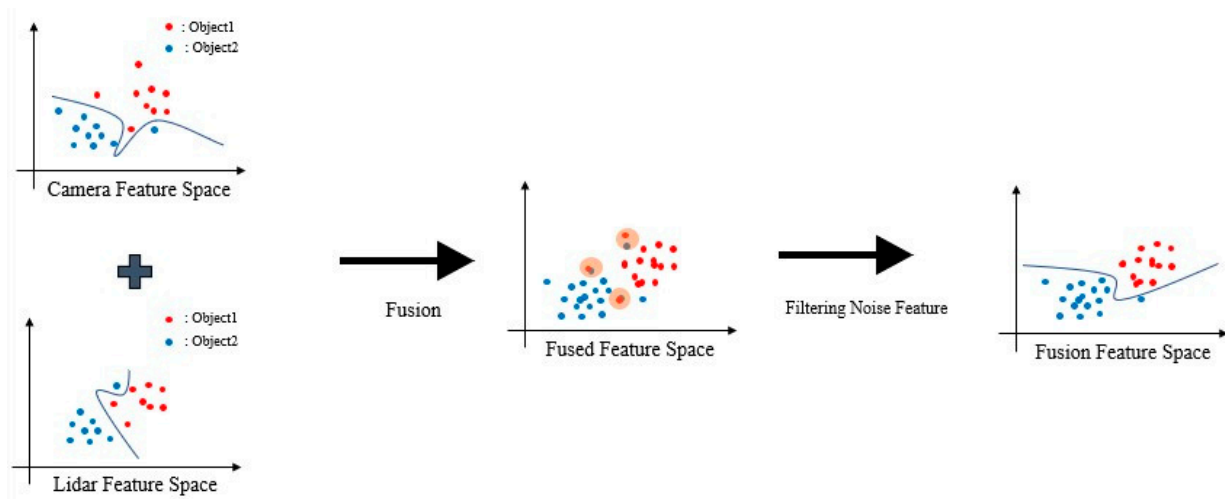


Figure 1. A noise feature occurs when fusion is simply merged in feature space. This makes object detection difficult. Noise features tend to occur when both sensors are weak.

In [13], the authors proposed a feature recalibration method to filter the noisy features while training using data labels, also known as data annotations. However, annotations may not always be suitable for training the network to recognize various data distributions. This is because annotations may not reflect real-world conditions, such as weather or lighting, which can introduce noise to the data. Therefore, relying solely on annotations

can result in another source of uncertainty that can impede learning. In order to handle this problem, the proposed method performs self-learning that utilizes robust features extracted from the network to overcome the limitations of annotation and improve the network's performance in recognizing objects under various conditions.

The proposed AFAM-EfficientDet utilizes four EfficientNet backbones for visual and lidar feature extraction consisting of two pairs known as the source and target networks. Both the network pairs differ in the training data. For each network, the lidar point cloud is converted into the dense range image before feature extraction. For efficient convergence, the extracted visual-lidar features are converted into query, key, and value to estimate the correlation between lidar features and their relevant camera features. Given the query, key, and value from the source and target networks, the AFAM first computes the uncertainty between the lidar features of the source and target networks and the camera features of the source and target networks. Based on the computed uncertainty, the AFAM adaptively computes the attention maps along the channel and spatial axis. The extracted camera and lidar features from the target network are recalibrated by element-wise multiplication with attention maps. Finally, the refined camera and lidar features are fused and given as input to the EfficientDet-B3 for object detection. We evaluated the performance of the proposed method for object detection in adverse weather conditions using the Dense Dataset [12]. To conduct the evaluation, we employed EfficientDet [14] and trained it five times. The robustness of the network's performance was assessed by calculating the difference between the maximum and minimum mean average precision (mAP) and by computing the average and deviation values. The experimental results demonstrate that our proposed method effectively improves the sensor fusion performance of object detection networks in adverse weather conditions, enabling them to operate more robustly in real-world scenarios.

Our main contributions are as follows:

1. We propose an effective adaptive feature attention module (AFAM) that can be widely applied to boost the representation power of CNNs.
2. We validate the effectiveness of our AFAM via ablation studies.
3. We verify that the AFAM outperforms the benchmark network EfficientDet on the benchmark dataset, the Dense Dataset.

The rest of the paper is structured as follows: In Section 2, a literature review of existing camera- and lidar-based object detection methods is presented. Section 3 describes the proposed adaptive feature attention module for the visual-lidar sensor fusion network in detail. Section 4 outlines the experimental setup and the results obtained. Finally, Section 5 concludes this research.

2. Related Work

In this section, we have discussed previous works related to object detection using camera and lidar sensors. Based on the input sensor data used for object detection, the existing literature can be grouped into three main categories: camera-based object detection methods, lidar-based object detection methods, and visual-lidar based object detection methods. The literature related to each of these categories is explained in the subsequent sections.

2.1. Camera-Based Object Detection

Camera-based object detection techniques, such as Fast-RCNN [15,16] and YOLO [17], have been advanced with the aid of contextual information provided in images to detect objects of varying sizes. However, the incorporation of rich contexts poses a challenge to the network training process. To address this problem, feature compression techniques, like squeeze-and-excitation networks [18], and attention methods, such as CBAM [19], have been proposed. In low-visibility conditions, such as fog and darkness, object detection becomes a challenging task. Existing methods employ dehazing techniques for object detection in bad weather and illumination conditions where dim images are transformed into brighter ones, thus enabling better detection [20–25]. Though dehazing methods

show improved performance, they require the same scene in clear weather conditions, necessitating the use of synthetic data. Moreover, computational efficiency is another challenge for camera-based object detection methods. Recently, transformer-based object detection methods [26–29] have emerged as a solution to address the computational power required by such methods [28]. Nonetheless, such methods have been crucial to improving object detection performance, particularly in challenging weather conditions and lighting conditions.

2.2. Lidar-Based Object Detection

On the other hand, extensive research has been performed to analyze the impact of weather conditions such as fog and high humidity on LiDAR sensor data for object detection problems [30–32]. Heinzler et al. [30] artificially induced humidity in a chamber and examined how the data measurements of human and vehicle objects were affected under foggy conditions. It was found that humidity has a significant impact on the distribution of LiDAR data. Object detection using LiDAR data is typically achieved by Point-Net [33,34] and the Voxel-based Network [35–37]. PointNet is designed to learn consistent features of LiDAR data, but it struggles to identify invariant features when there is noise in the data. On the other hand, Voxel-based object detection suffers from unnecessary data in the grid, which makes network training difficult, especially in high humidity conditions. Moreover, lidar-based object detection methods suffer from the point cloud sparse distribution in 3D space, which affects the detection performance. Motivated from the benefits of camera-based object detection, LaserNet [38] generates cylindrical range images using lidar data, allowing for more effective noise removal and greater contextual information extraction via convolution. Moreover, utilizing dense images enables more information to be extracted from the CNN kernel even for the regions where the point cloud is sparse. This method was successful in achieving high detection performance on significantly large dataset; however, its performance degrades when training is insufficient.

2.3. Visual–Lidar-Based Object Detection

In the past few years, many multi-sensor fusion-based object detection methods have been proposed to overcome the limitations of a single modality, i.e., camera or lidar [4–7,9,12–14,39]. The existing fusion architectures can be grouped into three main categories based on the stage at which they merge features from different modalities: early fusion, deep fusion, and late fusion [6]. In early fusion, data from different modalities are combined at the input stage [5,40]. Deep fusion utilizes distinct networks for different modalities and simultaneously integrates intermediate features [4,5,7,11]. Late-fusion-based methods handle each modality separately and merge their outputs at the decision-making level [6,41,42].

The use of multisensory fusion-based methods can lead to good performance, but their effectiveness may decrease when one of the sensors fails to function properly in adverse weather conditions. To address this issue, researchers have proposed the feature switch layer [13] and FIFO [43] to teach distinctive features that are specific to the current environmental conditions. This enables the robust fusion of multisensory data in challenging weather conditions such as fog and low light. However, relying solely on dataset annotations for training the network may not always be ideal as real-world conditions such as weather and lighting can introduce noise to the data, making the annotations less suitable. This could lead to uncertainty and hinder learning.

Self-supervised learning-based object detection methods can be solutions for such problems [44]. Contrastive learning is utilized to assess uncertainty by evaluating learned and unlearned data. Uncertainty, in this context, pertains to the ability of the network to determine the similarity between the learned data and new data that need to be learned, allowing for an assessment of the data distribution [45].

In this research, we focus on visual–lidar fusion-based object detection and develop the adaptive feature attention module for the deep fusion of extracted features for adverse weather conditions.

3. Proposed Method

In Section 3.1, we first present the overall multisensory deep fusion network pipeline. Then, we explain the proposed adaptive feature attention module (AFAM) in Section 3.2. Finally, we explain the training of the object detection network with the AFAM in Section 3.3.

3.1. Network Pipeline

Figure 2 illustrates the working pipeline of the proposed AFAM–EfficientDet network. The network takes lidar and image data as input and generates the prediction scores and semantic labels for the detected objects in the current view.

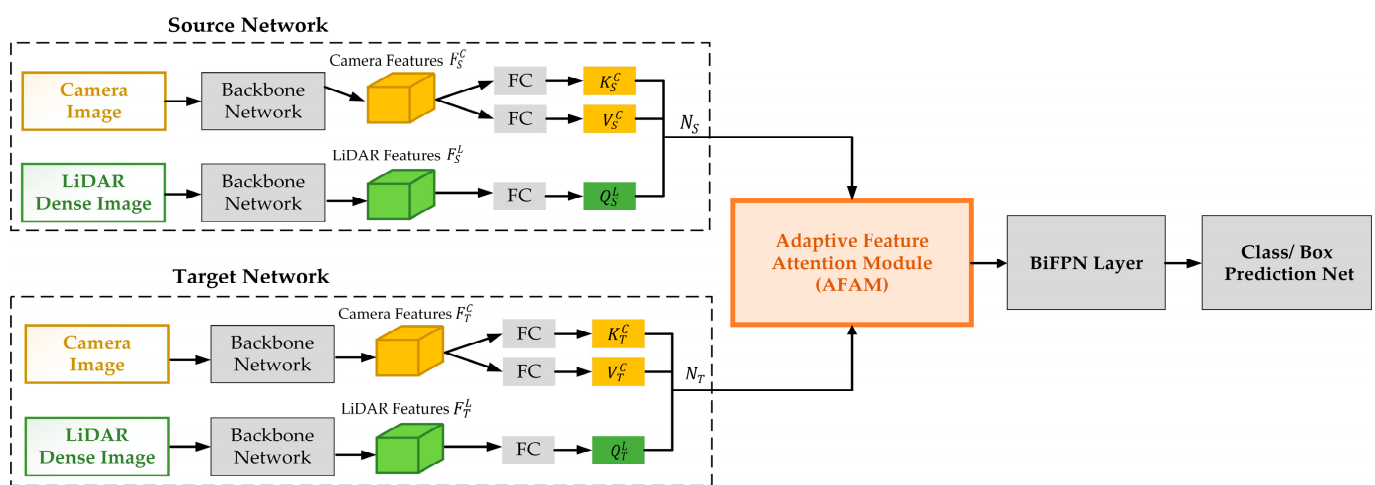


Figure 2. The network pipeline of the proposed AFAM–EfficientDet network.

The proposed method uses EfficientNet [46] as a backbone for feature extraction from camera and LiDAR data. The lidar data exhibit inherent density when observed from the sensor’s perspective, but it becomes sparser upon projection into a 3D space. This sparsity arises due to the constant angular density of the measurements, resulting in a larger number of measurements for objects in close proximity compared to those located further away. Moreover, the coordinate system of the raw lidar point cloud differs from the camera coordinates, which makes visual–lidar fusion-based object detection a complex problem. In order to deal with the aforementioned problem, this research generates a dense range image from the raw lidar data as performed in [38]. The obtained dense image is the range view representation of lidar data and is obtained by projecting the point cloud onto a camera coordinate system. It comprises three channels: depth, height, and intensity. The dense image offers a denser point cloud, allowing the use of a convolutional neural network (CNN) kernel size equivalent to that of a camera. This facilitates the efficient alignment of coordinate systems across different sensors, enabling effective convergence. The proposed method leverages the converted lidar and camera data to achieve its objectives.

Thus, four backbone networks [35] consisting of two pairs known as the source and target networks are utilized. Each pair comprises a camera and a lidar sensor backbone network. The key distinction between the source and target networks lies in the configuration of their respective training datasets. The source network is trained solely on camera images and LiDAR point cloud data captured during daytime and clear weather conditions [12], while the target network is trained using data captured during clear weather, adverse weather, and illumination conditions. The features F extracted from the source S and

target T network given as F_S and F_T are expressed as a set of camera C and lidar L features extracted from the source and target networks, as illustrated in Equations (1) and (2):

$$F_S = \{F_S^C, F_S^L\} \quad (1)$$

$$F_T = \{F_T^C, F_T^L\} \quad (2)$$

where, F_S^C and F_T^C represents the n camera features extracted from the source and target backbone networks given in Equation (3), while F_S^L and F_T^L denote the m Lidar features extracted from the source and target backbones, as depicted in Equation (4). Thus, features F are of size $W \times H \times D$, where W (width) and H (height) are the spatial dimensions, and D (depth) is the number of channels extracted from the backbone network.

$$F^C = \{C_1, C_2, \dots, C_n\} \quad (3)$$

$$F^L = \{L_1, L_2, \dots, L_m\} \quad (4)$$

For the efficient convergence of lidar features with their relevant camera features, we employ a cross-attention mechanism [9] that captures correlations between the two modalities in a dynamic manner. The input consists of a voxel cell and its corresponding N camera features. By utilizing three fully connected layers, we individually transform the voxel into a query Q^L and the camera features into key K^C and value V^C vectors. The inner product operation is then applied between the query and keys, resulting in an attention affinity matrix. This matrix encapsulates the $1 \times N$ correlations between the voxel and its associated camera features. To ensure proper weighting, the attention affinity matrix is normalized using a SoftMax operator. Subsequently, this normalized matrix is used to weigh and aggregate the camera feature values V^C , which contain relevant camera information. The resultant feature vectors for source and target networks N_S and N_T with the corresponding query Q^L , key K^C , and value V^C , depicted in Equations (5) and (6), are given as input to the adaptive feature attention module (AFAM) for the adaptive learning and recalibration of the features based on the computed uncertainty, as explained in Section 3.2:

$$N_S = \{Q_S^L, K_S^C, V_S^C\} \quad (5)$$

$$N_T = \{Q_T^L, K_T^C, V_T^C\} \quad (6)$$

The AFAM module outputs the fused camera–lidar features, which are given as input to BiFPN [14] for fast multi-scale feature fusion. The fused features are fed to the object class and the box detection head. The BiFPN and detection head are configured following the EfficientDet-B3 [14].

3.2. Adaptive Feature Attention Module (AFAM)

The AFAM takes N_S and N_T as input and compares them to learn the robust features for object detection in adverse weather and illumination conditions. The major components of AFAM include uncertainty computation, adaptive channel attention, and spatial attention, as shown in Figure 3.

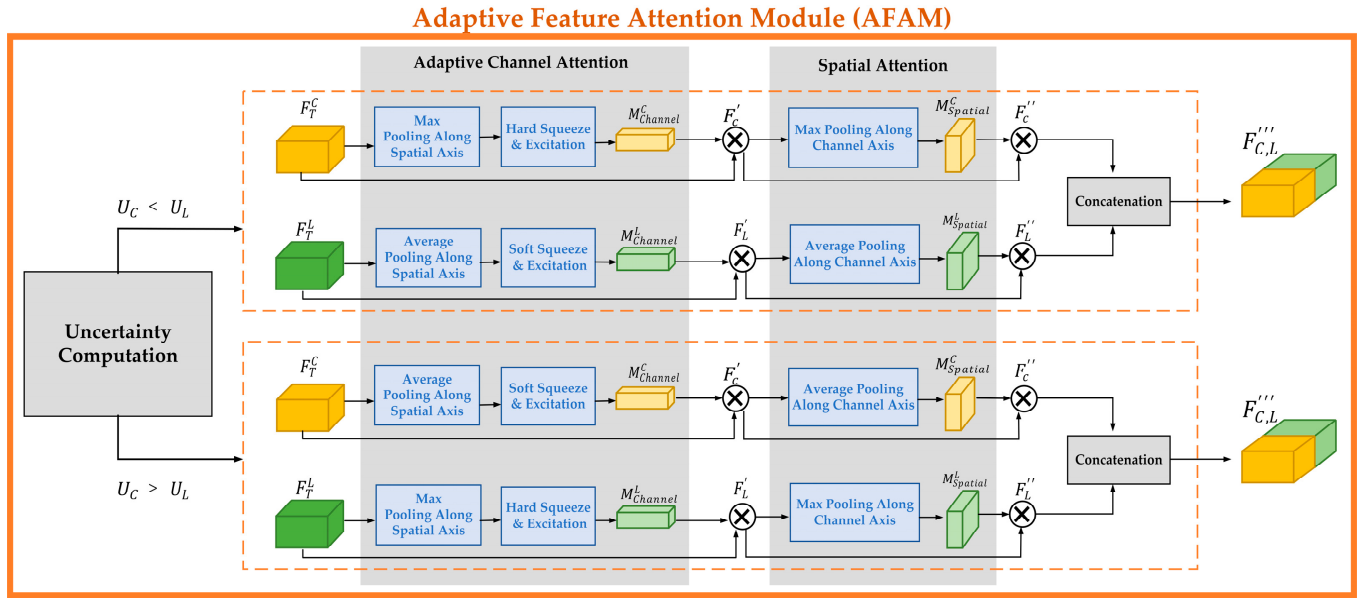


Figure 3. Overview of AFAM. The module has three sequential sub-modules: uncertainty computation, adaptive channel, and spatial attention. The module is embedded with EfficientDet-B3. The extracted camera and lidar features are refined and fused using our model AFAM.

3.2.1. Uncertainty Computation

The uncertainty is computed by comparing the query, key, and value of the source S and target T networks with each other. Let Q_S^L , in Equation (5), have n number of queries denoted as $q_S^{1...n}$ with their corresponding key and value represented as $k_S^{1...n}$ and $v_S^{1...n}$, respectively. On the other hand, let Q_T^L , in Equation (6), have m number of queries with their corresponding key and value given as $q_T^{1...m}$, $k_T^{1...m}$, and $v_T^{1...m}$. In this scenario, each i th query from the source network q_S^i , where $1 \leq i \leq n$, is compared with the j th query received from the target network q_T^j with $1 \leq j \leq m$. This comparison is performed to determine the similarity S' between the q_S^i and q_T^j , denoted as S'_{Q_i} when $i = j$ using Equation (7):

$$S'_{Q_i}(q_S^i, q_T^j) = \frac{q_S^i \cdot q_T^j}{\|q_S^i\| * \|q_T^j\|} \quad (7)$$

The variables i and j refers to the indexes of the query in Q_S^L and Q_T^L . Similarly, the similarity for key S'_K and value S'_V of the source and target networks is obtained using Equations (8) and (9) with i and j as indexes in the key and value arrays of the source and target networks:

$$S'_{K_i}(k_S^i, k_T^j) = \frac{k_S^i \cdot k_T^j}{\|k_S^i\| * \|k_T^j\|} \quad (8)$$

$$S'_{V_i}(v_S^i, v_T^j) = \frac{v_S^i \cdot v_T^j}{\|v_S^i\| * \|v_T^j\|} \quad (9)$$

The computed similarities $S'_{Q_{i...n}}$, $S'_{K_{i...n}}$, and $S'_{V_{i...n}}$ for the query, key, and value are summed to obtain the overall similarity S'_Q , S'_K and S'_V between the lidar data of the source and target networks using Equations (10) and (11):

$$S'_Q = \sum_{i=1}^n S'_{Q_i} \quad (10)$$

$$S'_K = \sum_{i=1}^n S'_{K_i} \quad (11)$$

$$S'_V = \sum_{i=1}^n S'_{V_i} \quad (12)$$

The similarity S'_Q , S'_K and S'_V between the source and target networks is used to compute the uncertainty of camera and lidar sensor data. For this purpose, the S'_K and S'_V are averaged to represent the combined similarity S'_C of the camera data, depicted in Equation (13). Equations (14) and (15) are used to compute the uncertainty for lidar U_L and camera U_C :

$$S'_C = \frac{1}{2} (S'_K + S'_V) \quad (13)$$

$$U_L = 1 - \frac{S'_Q}{S'_Q + S'_C} \quad (14)$$

$$U_C = 1 - \frac{S'_C}{S'_Q + S'_C} \quad (15)$$

3.2.2. Adaptive Channel Attention

The adaptive channel attention module takes uncertainty values U_L , U_C , and camera and lidar features of the target network $F_T = \{F_T^C, F_T^L\}$ as input and applies channel attention on the target network features to obtain the refined features for object detection, as shown in Figure 3. Channel attention is applied to the camera and lidar features given the following conditions:

- Case 1: $U_L < U_C$, the lidar data uncertainty is lower than the camera uncertainty. In such condition, max pooling is applied on the lidar features F_T^L , while camera features F_T^C are average pooled.
- Case 2: $U_C < U_L$, the similarity between camera features is higher in comparison to the lidar features of the source and target networks. In this case, max pooling is applied on the camera features F_T^C , while lidar features F_T^L are average pooled.

The rationale behind applying max pooling to features from sensors with low uncertainty is that regions in the feature vector with high values indicate a higher likelihood of object presence. Hence, when a sensor has low uncertainty, its extracted features are deemed more reliable, and max pooling is employed. Conversely, for sensors with high uncertainty, average pooling is applied to their features. This is because feature vectors extracted from uncertain sensors are expected to contain more noise, and averaging is used to filter out such noise. In situations where noise is prominent, it is common practice to employ averaging or outlier detection for data filtering.

After aggregating the spatial information of feature maps using adaptive max and average pooling, the squeeze-and-excitation [18] method is applied to dynamically recalibrate the channel-wise feature responses. This process aims to extract distinctive features while suppressing less informative ones. Specifically, to enhance features from sensors with low uncertainty, a higher compression ratio, referred to as “hard squeeze,” is employed. This higher compression ratio helps preserve robust features, allowing the network to focus on effectively learning them. Conversely, for sensor data with high uncertainty, average pooling is applied. Applying a high compression ratio to such data would lead to a significant reduction in feature values, resulting in decreased object detection performance. Hence, a “soft squeeze and excitation” approach with a lower compression ratio is utilized for sensor data with higher uncertainty.

The channel attention outputs the 1D channel attention maps, M^c and M^L , for camera and lidar features. Each map is of size $1 \times 1 \times D$, where D is the channel depth. The atten-

tion map is merged with the input features, F_T^C and F_T^L , using element-wise multiplication generating the refined features F_c' and F_L' , as given in the Equations (16) and (17):

$$F_c' = M_{Channel}^C(F_T^C) \otimes F_T^C \quad (16)$$

$$F_L' = M_{Channel}^L(F_T^L) \otimes F_T^L \quad (17)$$

3.2.3. Adaptive Spatial Attention

In order to capture the interspatial relationships of features, a spatial attention map is generated. This spatial attention differs from channel attention as it focuses on determining the informative regions. To compute the spatial attention, we perform average-pooling and max-pooling operations along the channel axis on the refined features F_c' and F_L' . The obtained attention spatial maps for camera and lidar features, $M_{Spatial}^C(F_c')$ and $M_{Spatial}^L(F_L')$, are merged with the input refined features using Equations (18) and (19) resulting in efficient feature descriptors F_c'' and F_L'' :

$$F_c'' = M_{Spatial}^C(F_c') \otimes F_c' \quad (18)$$

$$F_L'' = M_{Spatial}^L(F_L') \otimes F_L' \quad (19)$$

The obtained features F_c'' and F_L'' are concatenated, resulting in fused robust visual–lidar features $F_{C,L}'''$.

3.3. Training with AFAM

This subsection presents the training process of the object detection network using AFAM–EfficientDet. Table 1 enlists dataset traverses exhibiting different weather and illumination conditions and a number of samples used for training, testing, and validation from each of the traverses.

Table 1. Dataset size used for training, testing, and validation.

Dataset Traverse	Environmental Condition (Light, Weather)	Training	Validation	Testing
T ₁	Daytime, Clear	2183	399	1005
T ₂	Daytime, Snow	1615	226	452
T ₃	Daytime, Fog	525	69	140
T ₄	Nighttime, Clear	1343	409	877
T ₅	Nighttime, Snow	1720	240	480
T ₆	Nighttime, Fog	525	69	140
Total		8238	1531	3189

Algorithm 1 illustrates the overall training process using AFAM–EfficientDet. Firstly, the source and target networks are trained using T₁, which consists of data captured during daytime and clear weather. Randomized initial weights are used for each of the backbone. The training continues as long as the obtained loss $Loss_{total}$ is above the threshold ϵ . The value of ϵ is the same as in [13]. If $Loss_{total}$ falls below the threshold ϵ , the network starts training with the AFAM module.

Algorithm 1: Object Detection Network Training with AFAM–EfficientDet

Input: Camera Images and LiDAR dense range images
Output: List_of_class_predictions
 $E_{start} \leftarrow 0$
epoch $\leftarrow 0$
initialize random weights $W_S^C, W_S^L, W_T^C, W_T^L$
 $\epsilon \leftarrow 0.3$
while epoch $\leftarrow \infty$ **do**
 for each i in T_1 **do**
 input camera and LiDAR data into each backbone network // four backbones
 $F_S, F_T \leftarrow \text{feature_extraction}()$
 training source network
 training target network
 compute $Loss_{total}$
 if $Loss_{total} < \epsilon$ **then**
 $E_{start} \leftarrow \text{epoch}$
 break
 end if
 class_predictions_without_AFAM(i) \leftarrow predictions(labels, bounding_box, probability)
 $i++$
 end for
 epoch++
end while
List_of_class_predictions.append(class_predictions_without_AFAM)
epoch $\leftarrow 0$
while epoch $\leftarrow E_{start}$ **do**
 for j in T_{2-6} **do**
 input camera and LiDAR data into each backbone network // four backbones
 extract features from source and target network
 Refined_features \leftarrow Feature_Recalibration_with_AFAM(N_S, N_T, F_T)
 training target network // update weights of target network
 class_predictions_with_AFAM(j) \leftarrow predictions(labels, bounding_box, probability)
 $j++$
 end for
 epoch++
end while
List_of_class_predictions.append(class_predictions_with_AFAM)
return List_of_class_predictions

In case of $Loss_{total} < \epsilon$, the backbone networks are given camera and lidar data from T_{2-6} . The extracted features from the source and target networks are given as input to the AFAM. Based on the computed uncertainty, the target network's features F_T are refined. The target network is trained using the recalibrated features. The feature recalibration with AFAM is given in Algorithm 2.

Algorithm 2: Feature_Recalibration_with_AFAM

Input: N_S, N_T, F_T
Output: Refined Features $F_{C,L}'''$.
 $S'_Q, S'_K, S'_V \leftarrow$ Compute similarity(N_S, N_T)
 $U_C, U_L \leftarrow$ Compute uncertainty(S'_Q, S'_K, S'_V)
if $U_C < U_L$ **then**
 // camera feature recalibration
 $Pooled_F_T^C \leftarrow \text{maxpooling}(F_T^C)$
 $M_{Channel}^C(F_T^C) \leftarrow \text{hard_squeeze\&excitation}(Pooled_F_T^C)$

```


$$F'_c \leftarrow M_{Channel}^C(F_T^C) \otimes F_T^C$$


$$F''_c \leftarrow M_{Spatial}^C(F'_c) \otimes F'_c$$

//lidar feature recalibration

$$Pooled\_F_T^L \leftarrow \text{average\_pooling}(F_T^L)$$


$$M_{Channel}^L(F_T^L) \leftarrow \text{soft\_squeeze\&excitation}(Pooled\_F_T^L)$$


$$F'_L \leftarrow M_{Channel}^L(F_T^L) \otimes F_T^L$$


$$F''_L \leftarrow M_{Spatial}^L(F'_L) \otimes F'_L$$

else
//camera feature recalibration

$$Pooled\_F_T^C \leftarrow \text{averagepooling}(F_T^C)$$


$$M_{Channel}^C(F_T^C) \leftarrow \text{soft\_squeeze\&excitation}(Pooled\_F_T^C)$$


$$F'_c \leftarrow M_{Channel}^C(F_T^C) \otimes F_T^C$$


$$F''_c \leftarrow M_{Spatial}^C(F'_c) \otimes F'_c$$

//lidar feature recalibration

$$Pooled\_F_T^L \leftarrow \text{maxpooling}(F_T^L)$$


$$M_{Channel}^L(F_T^L) \leftarrow \text{hard\_squeeze\&excitation}(Pooled\_F_T^L)$$


$$F'_L \leftarrow M_{Channel}^L(F_T^L) \otimes F_T^L$$


$$F''_L \leftarrow M_{Spatial}^L(F'_L) \otimes F'_L$$

end if
Return  $F_{C,L}''' \leftarrow \text{concatenate}(F''_c, F''_L)$ 

```

4. Experiments and Results

4.1. Implementation Setup, Dataset, and Evaluation Parameters

This section discusses the experiments performed to evaluate the performance of the proposed network. All experiments are carried out on Intel core i7-9700, NVIDIA GeForce RTX 3080 using PyTorch library.

Addressing the object detection problem in adverse weather conditions and light changes, the proposed method is evaluated on publicly available benchmark dataset, i.e., the Dense Dataset [12]. This dataset is captured using a stereo camera, Velodyne 64ch LiDAR, and a radar exhibiting extreme light changes from day to night and weather conditions including clear weather, fog, and snow.

The open-source implementation of EfficientNet [47] is utilized as a backbone. The proposed method was employed using the AFAM when the total loss value reached a particular threshold. The critical value of AFAM learning was determined by applying the algorithm when the total loss value was 0.5. During the training process, the images were resized to a width and height of 896×896 pixels. The training was carried out for a maximum of 30 epochs, and the best validation dataset performance was used to determine the final model. The object detection network is trained for two semantic classes: pedestrian and vehicle.

The performance is evaluated using mean average precision (mAP). We have applied the PASCAL VOC 11-point interpolation method to compute the average precision (AP) for each class. Later, the average is computed using mean average precision across all the classes. In our case, there are two object class labels, i.e., pedestrian and vehicle. So, we compute the AP for each class using Equation (20):

$$AP_{label(i)} = \frac{1}{11} \sum_{r \in \{0, 0.1, \dots, 1\}} P_{interp}(r) \quad (20)$$

where $label = \{vehicle, pedestrian\}$, i is the index of class values in $label$, and P corresponds to the precision at each interpolated recall r . The mAP is computed using Equation (21). The IoU threshold was set to 0.5.

$$mAP_{0.5} = \frac{1}{n}(AP_{Pedestrian} + AP_{Vehicle}) \quad (21)$$

Here, n is the number of class labels.

4.2. Ablation Study

Table 2 displays the experimental outcomes of varying the configuration of the AFAM and assessing the performance based on query (Q), key (K), and value (V) applications. The paper posits that max pooling is effective in extracting appropriate features when there is a considerable shift in data, while avg pooling is optimal when there is minimal variation in data distribution. The table presents the results of experimenting with the configuration of Q when the LiDAR uncertainty value is high. The results indicate that using max pooling leads to better performance than using average pooling when the overall uncertainty is high. Additionally, the table lists the effect of using hard and soft squeeze with max and average pooling. It can be observed that incorporating squeeze and excitation with a relatively small ratio, soft squeeze, to the max pooling yields less information loss. In summary, the experimental results support the notion that the AFAM module configuration is reasonable.

Table 2. Different configurations used for AFAM module.

Network	Max Pooling	Average Pooling	Hard Squeeze	Soft Squeeze	Top1-mAP
AFAM-EfficientDet	Q	K, V	K, V	Q	0.419
	Q	K, V	Q	K, V	0.414
	K, V	Q	Q	K, V	0.405
	K, V	Q	K, V	Q	0.397

To determine the optimal compression ratio of information, experiments were conducted, and the results are detailed in Table 3. The table presents the experimental outcomes of varying the ratio of squeeze and excitation given as R_{hard} for hard squeeze and excitation, while R_{soft} is used for soft squeeze and excitation. The best results were achieved when compressing the output channel by $10\times$ or $20\times$. This experiment highlights the challenge of finding the appropriate hyperparameters to effectively utilize the AFAM.

Table 3. Experimental results obtained for compression ratio.

Network	R_{hard}	R_{soft}	Top1-mAP
AFAM-EfficientDet	16	8	0.419
	16	12	0.406
	16	16	0.395
	24	8	0.412
	24	12	0.398

In Figure 4, it is demonstrated that cameras struggle to identify objects in foggy conditions. Although it is daytime, the data distribution of snow is dissimilar to that of clear sunny days, with clustering occurring differently around 80 values. In sunny weather, the values are more closely clustered around 80. However, it is evident that the data distribution has a significant variance at 80 in snowy conditions. In the case of fog, the data values are clustered around 120. Based on this data distribution, the camera and LiDAR data are challenging to be used for network training due to the changes in data distribution unless the input data of the network are refined first.

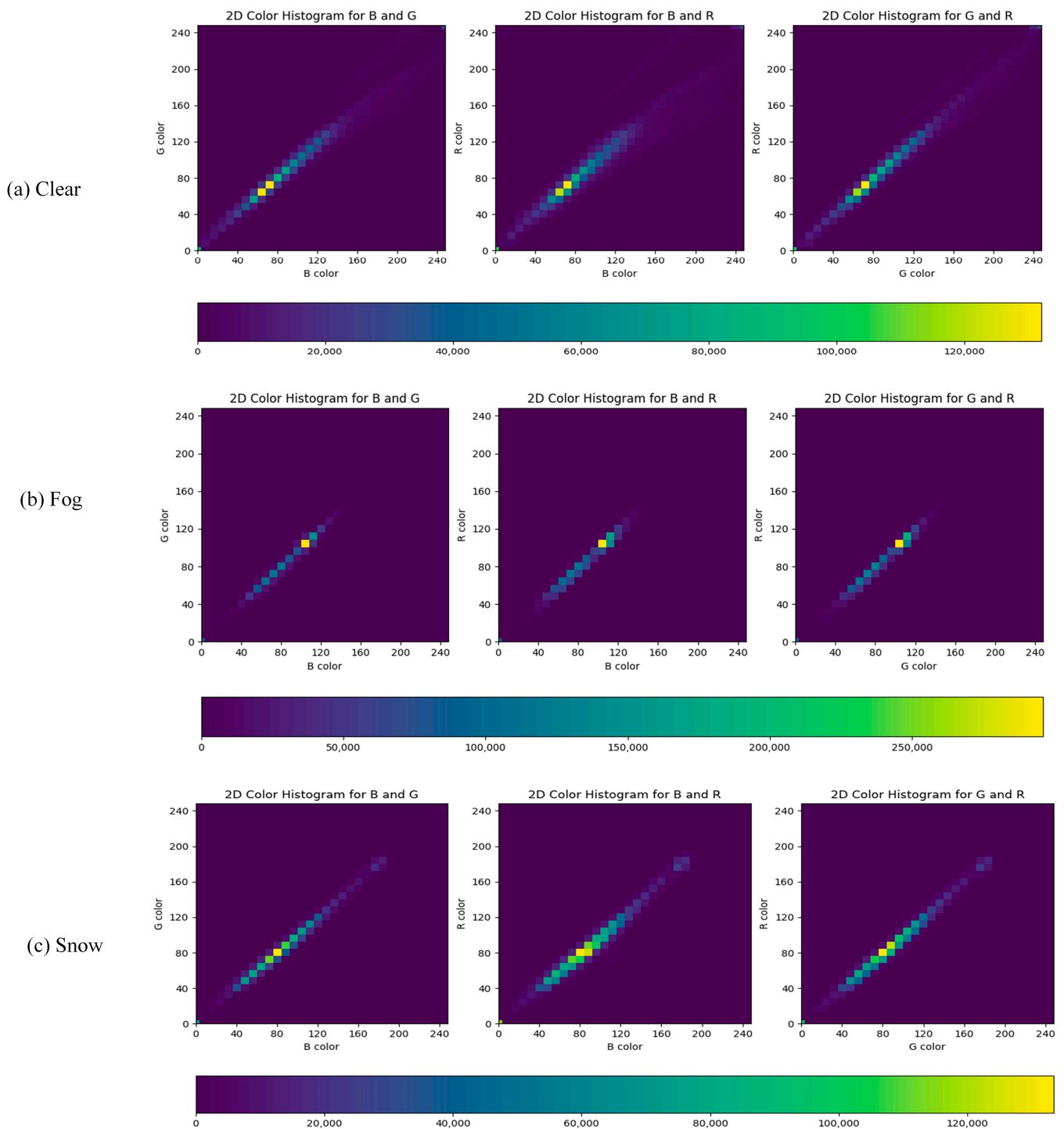


Figure 4. Histogram distribution based on weather for daytime data.

4.3. Comparison with Other Methods

This section presents the performance comparison of the proposed AFAM-EfficientDet with state-of-the-art object detection networks using single and multiple modalities, i.e., EfficientDet [14] using only camera sensor, EfficientDet with camera–lidar fusion, a Feature Switch Layer [13] enabled EfficientDet, and ResT [48] enabled EfficientDet. EfficientDet is typically built to optimize the network efficiency in terms of computational cost and robust feature fusion. Open-source implementation has been used for the implementation of EfficientDet. The feature switch layer (FSL) is designed for

object detection in adverse weather conditions. Based on the dynamic environmental conditions, the FSL extracts and fuses visual–lidar features for robust object detection. Both aforementioned methods and our proposed method use the same backbone network, which is EfficientNet [47]. ResT is used as a backbone network and is integrated with the EfficientDet for class prediction. For fair comparison, we have trained the backbone networks on the same dataset traverses T_{1-6} , as explained in Table 1, with their default configurations.

Table 4 lists the results obtained for four comparisons performed to evaluate the effectiveness of the proposed AFAM. The performance is evaluated based on the highest mean average precision (mAP) and variance recorded as a result of five training experiments. Top5-mAP depicts the mAP obtained for all five experiments, with Top1-mAP presenting the best mAP and Worst-mAP presenting the minimum mAP obtained among the five experiments. Variance is the difference between Top1-mAP and Worst-mAP.

Table 4. Performance comparison of proposed method with SOTA algorithms.

Comparison	Network	Modality	Top5-mAP	Top1-mAP	Worst-mAP	Variance
a	EfficientDet	C	0.347 ± 0.00073	0.367	0.318	0.049
	AFAM–EfficientDet	C	0.354 ± 0.00024	0.370	0.325	0.045
b	EfficientDet	C, L	0.398 ± 0.00018	0.414	0.377	0.037
	AFAM–EfficientDet	C, L	0.403 ± 0.00007	0.419	0.402	0.017
c	ResT–EfficientDet	C, L	0.234 ± 0.00232	0.247	0.205	0.042
	AFAM + ResT–EfficientDet	C, L	0.308 ± 0.00077	0.319	0.294	0.025
d	FSL	C, L	0.406 ± 0.00016	0.427	0.395	0.032
	AFAM	C, L	0.403 ± 0.00007	0.419	0.402	0.017

1. Comparison with baseline method (only camera features): Firstly, we compared the performance of the AFAM for camera only features. The EfficientDet takes raw features from the backbone network as input and predicts the object classes. On the other hand, AFAM when embedded with the EfficientDet performs the refinement of the features, thus providing more robustness for the object detection network. It is observed that using refined features enhances object detection performance.
2. Comparison with baseline method (visual–lidar fusion): Secondly, the performance of the multimodal EfficientDet that undergoes the deep fusion of camera and lidar features for object detection is analyzed and is compared with the AFAM–EfficientDet. It can be clearly seen that the AFAM, providing the more robust deep fusion of visual–lidar features, achieves a higher mAP in comparison to the multimodal EfficientDet.
3. Comparison with different network architecture: Here, we present the evaluation results when the AFAM is ported to different network architecture. For this purpose, first, we compute the results for ResT–EfficientDet. Here, ResT is used as a backbone network for visual–lidar feature extraction, and those features are given as input to EfficientDet for class prediction. To assess the effectiveness of the AFAM, we have replaced the backbone EfficientNet with ResT. The AFAM takes raw features from ResT as input, performs the feature recalibration and fusion, and then gives the fused visual–lidar features as input to the BiFPN layer of EfficientDet. As the backbone network is changed, the input features are different, resulting in a change in performance, which can be observed in Table 4. However, the AFAM–ResT–EfficientDet outperforms the ResT–EfficientDet.
4. Comparison with feature refinement method: Finally, we present the comparison of the AFAM with the FSL. Both the modules are used in integration with Efficient-

Det. They take the same features as input, perform the recalibration of the features, and output the refined visual–lidar features for class prediction. It is observed that FSL–EfficientDet achieves the highest Top1-mAP as it employs annotations for environment learning, while AFAM–EfficientDet adaptively learns via the environment’s dissimilar appearance and computes the uncertainty. In the case of dense fog, snow, and light changes from day to night, the camera–lidar-based object detection performance is significantly degraded. Based on the adaptive learning approach, AFAM–EfficientDet achieves the least variance, which is the difference between Top1-mAP and Worst-mAP, when the environment significantly changes due to illumination changes or adverse weather. On the other hand, annotation-dependent FSL–EfficientDet fails to deliver high performance under challenging weather conditions such as dense fog or extreme light changes, resulting in increased variance. Thus, the AFAM empowers the object detection network, EfficientDet, in this case, to achieve more robustness in adverse weather conditions.

Figure 5a–e illustrate the qualitative results of the object detection performed by the proposed AFAM–EfficientDet in comparison to EfficientDet using only camera features, EfficientDet using multimodal fusion, and FSL–EfficientDet. The results show that object detection performance is good during daytime in clear weather i.e., (Day, Clear). Using only the camera leads to decreased detection rates and non-detection in foggy conditions (Day, Fog), as given in Figure 5a. The fog at nighttime (Night, Fog) is even more challenging when camera features cannot be detected due to illumination variation. Object detection can be better achieved using multimodal fusion, while poor convergence resulted in the performance deterioration illustrated in Figure 5c. In contrast, visual–lidar fusion methods, shown in Figure 5d,e, have shown good detection performance even in fog and crowded situations, with the proposed method performing more robustly than FSL–EfficientDet. Moreover, the use of a multimodal fusion layer was found to enhance performance in all scenarios, surpassing the use of a single sensor. Interestingly, the network’s performance was observed to improve in foggy daytime conditions, indicating that it was compensating for the limitations of image-based fog detection. Figure 5f,g illustrate the qualitative results of the object detection performed by using ResT as backbone with EfficientDet. It is visible that when features are recalibrated using the AFAM, the performance of the ResT–EfficientDet is enhanced. These results ensure the portability of the AFAM.

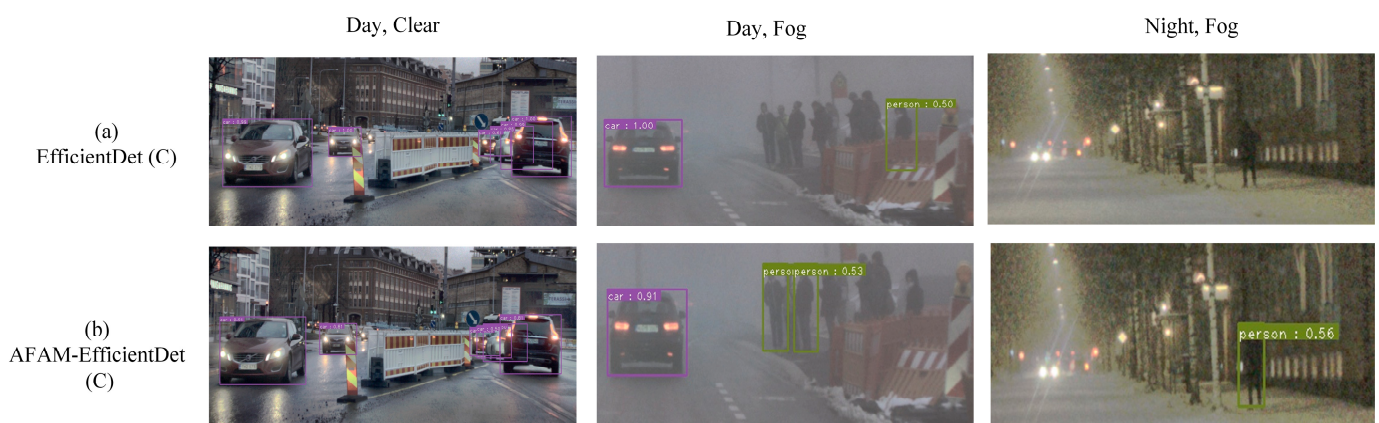


Figure 5. Cont.



Figure 5. Object detection performance qualitative results. (a,b) Only camera-feature-based object detection performance; (c–e) visual–lidar deep-fusion-based object detection performance; (f,g) integration of AFAM in different network architecture.

5. Conclusions

Object detection is crucial for autonomous navigation in dynamic environments. Extensive research has been performed in this field presenting single- and multi-sensor fusion-based object detection. However, adverse weather conditions and extreme illumination changes pose challenges for both camera and lidar sensors. This research presents a systematic study of the existing methods using cameras, lidar, and a fusion of both sensors for object detection. In order to address the shortcomings of previous literature, this research proposes an adaptive feature attention module (AFAM) that leverages the fusion of camera and lidar data and performs efficient object detection. The AFAM computes uncertainty between modalities and adaptively refines visual and lidar features using attention mechanisms along the channel and spatial axes. Integrated with the EfficientDet framework, the AFAM enhances object detection accuracy by effectively filtering noise and extracting discriminative information for object detection in specific environmental conditions. To evaluate the AFAM's effectiveness, we conducted experiments on a benchmark dataset that exhibits variations in weather and lighting conditions. The evaluation results demonstrate a significant improvement in the overall detection accuracy of the object detection network when the AFAM is employed, thus outperforming state-of-the-art methods. This research focuses on enhancing the performance of neural networks for object detection via sensor fusion, offering practical implications for real-world scenarios.

The AFAM contributes to the adaptive learning of the distinctive features for object detection in adverse weather conditions. In the future, we aim to extend this work for object detection in extreme seasonal changes along with varying weather and lighting

conditions and to evaluate benchmark datasets with diverse environments in order to generalize the model so that it can be applicable in any real-world environment. Furthermore, extending this research from static object detection to dynamic object tracking can be another future direction.

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

Funding: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2023-2020-0-01462) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

Data Availability Statement: Not applicable.

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

References

1. Khairuddin, A.R.; Talib, M.S.; Haron, H. Review on simultaneous localization and mapping (SLAM). In Proceedings of the 5th IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2015, Penang, Malaysia, 27–29 May 2016; pp. 85–90.
2. Arnold, E.; Al-Jarrah, O.Y.; Dianati, M.; Fallah, S.; Oxtoby, D.; Mouzakitis, A. A Survey on 3D Object Detection Methods for Autonomous Driving Applications. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 3782–3795. [\[CrossRef\]](#)
3. Kaur, J.; Singh, W. Tools, techniques, datasets and application areas for object detection in an image: A review. *Multimed. Tools Appl.* **2022**, *81*, 38297–38351. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Liang, M.; Yang, B.; Chen, Y.; Hu, R.; Urtasun, R. Multi-Task Multi-Sensor Fusion for 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7345–7353.
5. Ku, J.; Mozifian, M.; Lee, J.; Harakeh, A.; Waslander, S.L. Joint 3D Proposal Generation and Object Detection from View Aggregation. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Madrid, Spain, 1–5 October 2018; pp. 5750–5757.
6. Vora, S.; Lang, A.H.; Helou, B.; Beijbom, O. PointPainting: Sequential Fusion for 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 4604–4612.
7. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-View 3D Object Detection Network for Autonomous Driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1907–1915.
8. Huang, T.; Liu, Z.; Chen, X.; Bai, X. EPNet: Enhancing Point Features with Image Semantics for 3D Object Detection. In *Computer Vision—ECCV 2020*; Springer Science and Business Media Deutschland GmbH: Berlin/Heidelberg, Germany, 2020; pp. 35–52.
9. Li, Y.; Yu, A.W.; Meng, T.; Caine, B.; Ngiam, J.; Peng, D.; Shen, J.; Lu, Y.; Zhou, D.; Le Quoc, V.; et al. DeepFusion: Lidar-Camera Deep Fusion for Multi-Modal 3D Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 17182–17191.
10. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is All you Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
11. Xu, D.; Anguelov, D.; Jain, A. PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Madrid, Spain, 1–5 October 2018; pp. 244–253.
12. Bijelic, M.; Gruber, T.; Mannan, F.; Kraus, F.; Ritter, W.; Dietmayer, K.; Heide, F. Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11682–11692.
13. Kim, T.L.; Park, T.H. Camera-LiDAR Fusion Method with Feature Switch Layer for Object Detection Networks. *Sensors* **2022**, *22*, 7163. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [\[CrossRef\]](#)
16. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.

17. 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](#)]
18. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Madrid, Spain, 1–5 October 2018; pp. 7132–7141.
19. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
20. Patel, S.P.; Nakrani, M. A Review on Methods of Image Dehazing. *Int. J. Comput. Appl.* **2016**, *133*, 975–8887.
21. Zhang, Z.; Zhao, L.; Liu, Y.; Zhang, S.; Yang, J. Unified Density-Aware Image Dehazing and Object Detection in Real-World Hazy Scenes. In Proceedings of the Asian Conference on Computer Vision (ACCV), Kyoto, Japan, 30 November–4 December 2020.
22. Chen, W.-T.; Ding, J.-J.; Kuo, S.-Y. PMS-Net: Robust Haze Removal Based on Patch Map for Single Images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 11681–11689.
23. Berman, D.; Treibitz, T.; Avidan, S. Non-Local Image Dehazing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1674–1682.
24. Li, B.; Peng, X.; Wang, Z.; Xu, J.; Feng, D. AOD-Net: All-In-One Dehazing Network. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 4770–4778.
25. Liu, X.; Ma, Y.; Shi, Z.; Chen, J. GridDehazeNet: Attention-Based Multi-Scale Network for Image Dehazing. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7314–7323.
26. Zeng, C.; Kwong, S. Dual Swin-Transformer based Mutual Interactive Network for RGB-D Salient Object Detection. *arXiv* **2022**, arXiv:2206.03105.
27. Liu, Z.; Tan, Y.; He, Q.; Xiao, Y. SwinNet: Swin Transformer Drives Edge-Aware RGB-D and RGB-T Salient Object Detection. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 4486–4497. [[CrossRef](#)]
28. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 10012–10022.
29. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
30. Heinzler, R.; Schindler, P.; Seekircher, J.; Ritter, W.; Stork, W. Weather influence and classification with automotive lidar sensors. *IEEE Intell. Veh. Symp. Proc.* **2019**, *2019*, 1527–1534.
31. Sebastian, G.; Vatter, T.; Lukic, L.; Burg, C.; Schumann, T. RangeWeatherNet for LiDAR-only weather and road condition classification. *IEEE Intell. Veh. Symp. Proc.* **2021**, *2021*, 777–784.
32. Heinzler, R.; Piewak, F.; Schindler, P.; Stork, W. CNN-Based Lidar Point Cloud De-Noising in Adverse Weather. *IEEE Robot Autom. Lett.* **2020**, *5*, 2514–2521. [[CrossRef](#)]
33. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
34. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. PointPillars: Fast Encoders for Object Detection from Point Clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 12697–12705.
35. Zhou, Y.; Tuzel, O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Madrid, Spain, 1–5 October 2018; pp. 4490–4499.
36. Deng, J.; Shi, S.; Li, P.; Zhou, W.; Zhang, Y.; Li, H. Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 1201–1209. [[CrossRef](#)]
37. Mao, J.; Xue, Y.; Niu, M.; Bai, H.; Feng, J.; Liang, X.; Xu, H.; Xu, C. Voxel Transformer for 3D Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 3164–3173.
38. Meyer, G.P.; Laddha, A.; Kee, E.; Vallespi-Gonzalez, C.; Wellington, C.K. LaserNet: An Efficient Probabilistic 3D Object Detector for Autonomous Driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 12677–12686.
39. Pang, S.; Morris, D.; Radha, H. CLOCs: Camera-LiDAR Object Candidates Fusion for 3D Object Detection. *IEEE Int. Conf. Intell. Robot. Syst.* **2020**, 10386–10393. [[CrossRef](#)]
40. Cai, Z.; Fan, Q.; Feris, R.S.; Vasconcelos, N. A unified multi-scale deep convolutional neural network for fast object detection. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 354–370.
41. Hoffman, J.; Gupta, S.; Darrell, T. Learning with Side Information Through Modality Hallucination. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 826–834.
42. Song, S.; Xiao, J. Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 808–816.

43. Gsai, S.L.; Suha, T.A.; Gsai, K. FIFO: Learning Fog-Invariant Features for Foggy Scene Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 18911–18921.
44. Wu, Z.; Xiong, Y.; Yu, S.X.; Lin, D. Unsupervised Feature Learning via Non-Parametric Instance Discrimination. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Madrid, Spain, 1–5 October 2018; pp. 3733–3742.
45. Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; Com, W.; Deepmind, G. Weight Uncertainty in Neural Network. In Proceedings of the 32nd International Conference on Machine Learning, PMLR, Lille, France, 6–11 July 2015; pp. 1613–1622.
46. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
47. GitHub–lukemelas/EfficientNet-PyTorch: A PyTorch Implementation of EfficientNet and EfficientNetV2. Available online: <https://github.com/lukemelas/EfficientNet-PyTorch> (accessed on 1 May 2023).
48. Zhang, Q.-L.; Yang, Y.-B. ResT: An Efficient Transformer for Visual Recognition. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 15475–15485.

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.