

## Article

# Detection in Adverse Weather Conditions for Autonomous Vehicles via Deep Learning

Qasem Abu Al-Haija \* , Manaf Gharaibeh  and Ammar Odeh 

Department of Computer Science/Cybersecurity, Princess Sumaya University for Technology, Amman 11941, Jordan; m.gharaibeh@psut.edu.jo (M.G.); a.odeh@psut.edu.jo (A.O.)

\* Correspondence: q.abualhaija@psut.edu.jo

**Abstract:** Weather detection systems (WDS) have an indispensable role in supporting the decisions of autonomous vehicles, especially in severe and adverse circumstances. With deep learning techniques, autonomous vehicles can effectively identify outdoor weather conditions and thus make appropriate decisions to easily adapt to new conditions and environments. This paper proposes a deep learning (DL)-based detection framework to categorize weather conditions for autonomous vehicles in adverse or normal situations. The proposed framework leverages the power of transfer learning techniques along with the powerful Nvidia GPU to characterize the performance of three deep convolutional neural networks (CNNs): SqueezeNet, ResNet-50, and EfficientNet. The developed models have been evaluated on two up-to-date weather imaging datasets, namely, DAWN2020 and MCWRD2018. The combined dataset has been used to provide six weather classes: cloudy, rainy, snowy, sandy, shine, and sunrise. Experimentally, all models demonstrated superior classification capacity, with the best experimental performance metrics recorded for the weather-detection-based ResNet-50 CNN model scoring 98.48%, 98.51%, and 98.41% for detection accuracy, precision, and sensitivity. In addition to this, a short detection time has been noted for the weather-detection-based ResNet-50 CNN model, involving an average of 5 (ms) for the time-per-inference step using the GPU component. Finally, comparison with other related state-of-art models showed the superiority of our model which improved the classification accuracy for the six weather conditions classifiers by a factor of 0.5–21%. Consequently, the proposed framework can be effectively implemented in real-time environments to provide decisions on demand for autonomous vehicles with quick, precise detection capacity.



**Citation:** Al-Haija, Q.A.; Gharaibeh, M.; Odeh, A. Detection in Adverse Weather Conditions for Autonomous Vehicles via Deep Learning. *AI* **2022**, *3*, 303–317. <https://doi.org/10.3390/ai3020019>

Academic Editor: Giovanni Diraco

Received: 15 March 2022

Accepted: 14 April 2022

Published: 18 April 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 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/>).

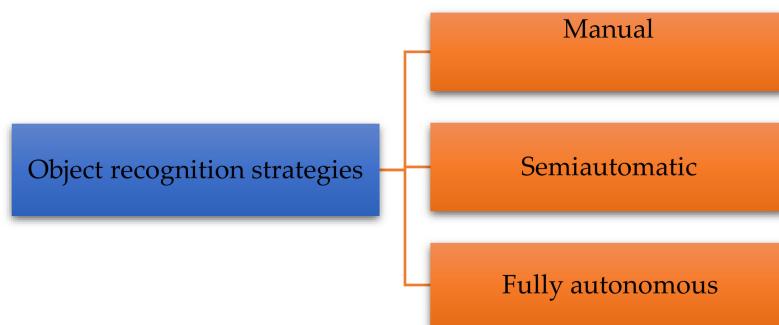
## 1. Introduction

Vehicle detection efficiency is a vital step in traffic monitoring and intelligent visual surveillance in general. The development of sensors and GPUs, as well as deep learning algorithms, has recently focused research on autonomous or self-driving applications based on artificial intelligence, which has become a trend [1,2]. To make the best control decisions and provide the required safety, autonomous cars must accurately recognize traffic objects in real time. Various sensors, including cameras and light detection and range, are often used in autonomous vehicles to detect such objects [3].

Adverse weather conditions, such as heavy fog, sleet rain, snowstorms, dusty blasts, and low light conditions, have a significant impact on the image quality of these various types of sensors [4,5].

As a result, visibility is insufficient for accurately detecting vehicles on the road, resulting in traffic accidents. Developing efficient image-enhancing approaches to obtain an excellent visual appearance or discriminative qualities will help achieve clear visibility [6,7]. As a result, supplying excellent images to detection systems can increase vehicle identification and tracking performance in intelligent visual surveillance systems and autonomous vehicle applications [8,9].

Several research and surveys have presented numerous object recognition strategies for in-vehicle situations, the three most frequent detection methodologies being manual, semiautomatic, and fully autonomous [10]. We summarize them in Figure 1. Manual and semiautomated surveys are the typical ways for gathering information about things found on roads. In a manual approach, a visual inspection of the objects present on the streets/roads is carried out either by walking or by driving a slow-moving vehicle along the streets/roads. Inspectors' subjective judgments are a problem in such inspections [11,12]. Given the length of road networks and the number of items, it necessitates significant human interaction, proven time-consuming. Furthermore, inspectors are frequently required to be physically present in the travel lane, putting them in potentially dangerous situations [13,14].



**Figure 1.** Object recognition strategies.

Objects on the roads/streets are collected automatically from a fast-moving vehicle in semiautomated object detection techniques, and the acquired data are processed on office workstations [15]. This strategy enhances safety, but it is still dependent on policy, which takes a long time to develop. Vehicles outfitted with high-resolution digital cameras and sensors are frequently used in fully automated object-detecting procedures [16].

Pretrained recognition-software-based models are used to detect automobiles and surrounding objects from gathered images/videos. Data processing might occur during data collection or afterward in the office as postprocessing. To capture road assets, specialized vehicles equipped with various sensors such as laser scanners and LiDAR (light detection and ranging) cameras are typically employed for automatic object detection. Vehicle-based traffic detection is commonplace since it allows for more efficient and faster object screening [17].

Different vehicle detection algorithms have recently been presented in the computer vision community. Because it provides high detection accuracy [18], deep-learning-based traffic object identification using video sensors has grown more critical in autonomous vehicles [19]. As a result, it has become the primary method in self-driving applications. The detector must meet two requirements: real-time detection is required for an active echo of vehicle controls, and high detection accuracy of traffic objects is needed, which has never been examined under severe weather conditions before [20].

Under adverse weather conditions, real-time detection is required for traffic monitoring and self-driving applications. Though real-time detection is possible, as in [18], it is difficult to use in severe weather settings. As a result, the preceding solutions are insufficient in terms of a trade-off between detection precision and detection time, which limits their use in applications involving lousy weather [21].

It is critical to use a vehicle detector with high accuracy and to consider this factor alongside detection speed to reduce false alarms of detected bounding boxes and to allow time to improve visibility in the traffic environment during inclement weather, thereby avoiding traffic accidents [22].

Compared to standard machine-learning-based approaches [23], deep neural networks have significantly enhanced the performance of intelligent autonomous or self-driving cars,

smart surveillance, and smart-city-based applications. Deep learning, which is based on neural networks, is a more advanced type of machine learning that can solve problems in various complicated application models that cannot be solved using typical statistical methods [24].

Therefore, in this paper, we make use of deep neural networks by a means of transfer learning techniques leveraging the power of the Nvidia GPU equipped by our high-performance commodity machine to provide weather detection in six types of weather conditions (cloudy, rainy, snowy, sandy, shine, and sunrise). Specifically, we develop a deep learning framework for weather detection that characterizes the performance of three known efficient deep convolutional neural networks (CNNs), namely, SqueezeNet CNN, ResNet-50 CNN, and EfficientNet-b0 CNN. The proposed model tends to be accurate, sensitive, precise, and lightweight to be efficiently deployed with autonomous vehicles in order to provide help for autonomous vehicles to make proper decisions, especially in adverse weather.

There are three basic steps in the suggested model. The data preparation, learning model, and evaluation subsystems were used to evaluate the system performance using different evaluation metrics and generate categorization for every weather image through a multiclass classifier. Based on the evaluation measures (accuracy, precision, sensitivity, and F1-score) [25], the simulation results demonstrated a high performance compared to previous models. Furthermore, the proposed model is considered the fastest in terms of time using medium-cost hardware, which helps to apply it widely in the system of the autonomous vehicles with high efficiency, which helps to preserve human lives and make decisions quickly with low-cost hardware (GPU) compared to other systems.

The rest of this paper is organized as follows: Section 2 reviews some of the main and recent related research work. Section 3 provides a comprehensive description about the developed model architecture. Section 4 reports on the system evaluation with extensive results and comparisons. Finally, Section 5 concludes the paper and provides some suggested future work.

## 2. Related Works

Taking advantage of the rapidly evolving deep learning techniques and deep CNN, automatic recognition of weather conditions from visual contents has received much attention from many research groups in recent years. This section discusses prior work related to recognizing weather conditions from images.

Like in our work, Elhoseiny et al. tackled the weather classification problem using convolutional neural networks (CNNs) [26]. This work also studied the feature space introduced at different CNN layers. Their work considered only a two-class weather classification task, namely sunny and cloudy classes. Their methodology uses fine-tuning, where they start from a pretrained network, which is then adapted to the weather classification. The proposed CNN architecture consists of five convolutional and pooling layers preceding three fully connected layers, including the final output layer. This CNN output layer has two nodes corresponding to the sunny and cloudy classes. For training and evaluation, a dataset with 10,000 images representing two classes was used. Their approach achieved 82.2% normalized classification accuracy (91.1% in regular classification accuracy).

Using cameras as weather sensors, Chu et al. proposed a model to estimate weather conditions from images [27]. In their model, they build a large image dataset of over 180,000 photos with weather properties, including weather type (sunny, cloudy, snowy, rainy, and foggy), temperature, and humidity. The authors filter out indoor photos by implementing a support vector machine (SVM) for indoor/outdoor classification, reporting 98% accuracy. They also filter out photos if the sky comprised less than 10% of the photo. They use geotags and time information associated with an image to collect related weather information from a web-based weather platform. To infer weather information for a given image, this work uses random forests to construct a computational model based on the correlation between weather properties and metadata. They reported a 58% average

accuracy of classifying weather types using different weather-related features. As an application for their model, they built a weather-aware landmark classification application that uses weather information as part of the classification process.

Motivated by the impact of severe weather conditions on urban traffic and the development of deep learning, Xia et al. proposed the simplified model, ResNet-15 [28]. The proposed model is based on the residual network ResNet-50, where layers are reduced from 50 to 15. Their approach is based on a deep CNN that extracts weather characteristics for elements such as the sky and the road using the convolutional layers. The input images are classified using the fully connected layers and SoftMax classifier. This simplified model was reported to work efficiently even on a standard CPU. This work also created a dataset of weather images on traffic road. The dataset consisted of four categories and contains around 5k weather-related images. The proposed system was trained and tested on this dataset. The authors considered four weather conditions and reported their recognition accuracy as follows: foggy (96.4%), rainy (97.3%), snowy (94.7%), and sunny (95.1%).

Ibrahim et al. introduced a framework to extract weather information from street-level images [29]. The authors' approach utilizes deep learning and computer vision using a unified method without predefined constraints in the used images. The model extracts various weather conditions at different times such as dawn/dusk, day, and night to enable time detection. This study proposes four deep CNN models to detect different visibility conditions, including dawn/dusk, day, night-time, glare, and weather conditions such as rain and snow. The reported recognition accuracy for these different classes ranges from 91% to 95.6%.

To classify weather conditions, Xiao et al. proposed MeteCNN, a deep CNN [30]. This study created a dataset of 6,877 images, which they labeled into 11 weather categories: hail, rainbow, snow, and rain. The categorization was based on visual shapes and color characteristics in the images. This dataset was used to train and test the proposed model of 13 convolutional layers, six pooling layers, and a SoftMax classifier. The classification accuracy of the proposed model over the testing set was reported around 92.

Earlier work by Roser et al. attempted to enhance driver assistance systems (DAS) on vehicles [31]. They developed histogram features for weather classification and constructed an SVM classifier based on contrast, intensity, sharpness, and color features to classify images captured by a camera mounted on a vehicle into different weather conditions such as clear, light rain, and heavy rain weather. The study reports accuracy results with around 5% error with carefully selected features for the SVM classifier.

Also motivated by the need to improve DAS and address the disruption of their vision-assisted functionality due to weather conditions such as haze and rain, Kang et al. proposed a deep-learning-based weather recognition framework [32]. The framework considers three common weather conditions: hazy, rainy, and snowy weather. The study experimented with well-known deep networks, GoogLeNet and AlexNet, to evaluate the proposed method after a few modifications to achieve the four output classes. The study also compared the deep learning methods against the hand-crafted feature-based method. The study results suggest that the deep learning methods are much better.

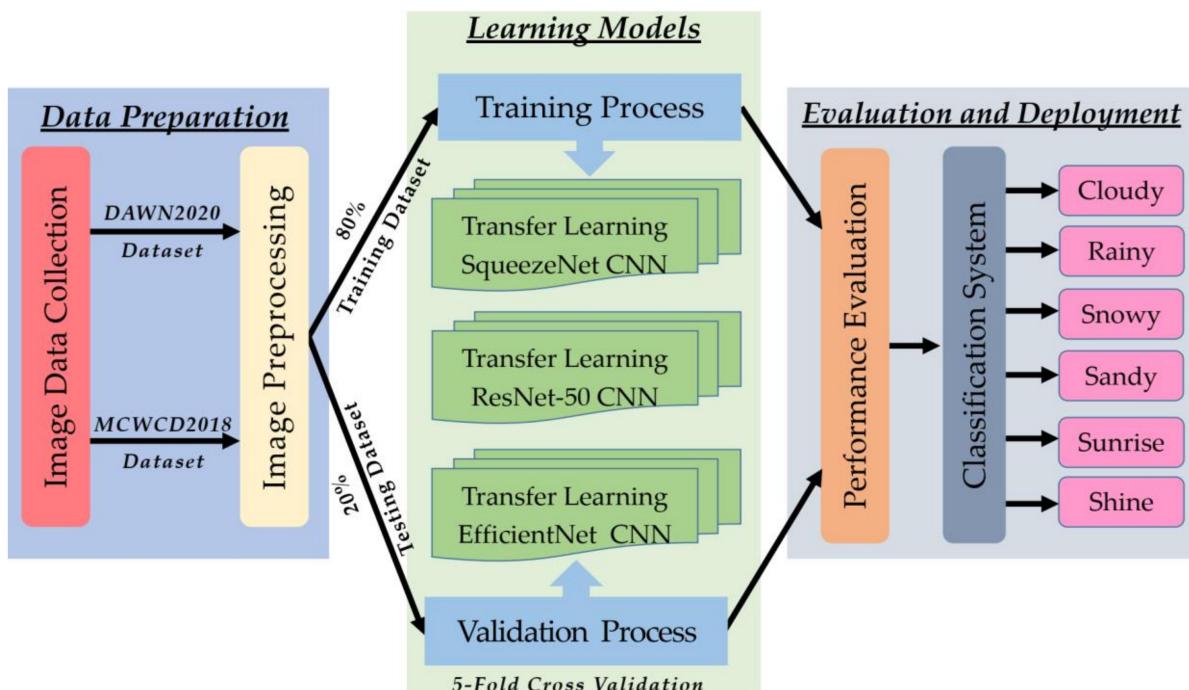
Guerra et al. created a new public dataset consisting of images for three weather classes, including rain, snow, and fog [33]. The study also proposed an algorithm that uses super pixel delimiting masks for data augmentation. The study examined different CNN models, such as CaffeNet, PlacesCNN, ResNet-50, and VGGNet16, to determine if any of them benefit more from the super pixel masks. The studied categories included sunny, cloudy, foggy, rainy, and snowy. Overall, the reported results for all models were between 68% and 81%, with the ResNet-50 being the most accurate overall.

Apart from traditional learning techniques, this work utilized the use of transfer learning techniques from the deep convolutional neural networks that are pretrained using the ImageNet dataset by applying fine-tuning at the input and the output layers while freezing the trainable parameters at the middle pretrained layers to improve the computational complexity of the deep models. On the other hand, leveraging the capability

of the Nvidia GPU provided by commodity machines, we trained the detection model using a combined dataset that could classify weather conditions into one of the six classes (cloudy, rainy, snowy, sandy, shine, and sunrise). Particularly, our transfer-learning-based system differentiated the key performance indicators, SqueezeNet CNN, ResNet-50 CNN, and EfficientNet-b0 CNN, and accordingly provided extensive experimental results and comparisons that eventually exhibited the superiority of our ResNet-50-based model over other implemented models (the SqueezeNet-based model and EfficientNet-b0-based model) and the state-of-the-art models as well.

### 3. DAWN Detection Model

The main goal of this research is the development of a system that can detect and classify weather conditions in adverse or normal situations using deep learning methods. The intended system is assumed to be deployed with autonomous vehicles and can effectively identify real-time outdoor weather conditions. Consequently, autonomous vehicles can make appropriate decisions and accordingly adapt to new conditions and environments. Specifically, our system of interest (SOI) is composed of three subsystems (illustrated in Figure 2): the data preparation (DP) subsystem to handle the collection and preprocessing of weather conditions image datasets; the learning models (LM) subsystem to train, validate, and test the input image datasets using deep learning (DL) algorithms; and the evaluation and deployment (ED) subsystem to evaluate system performance using different evaluation metrics and generate categorizations for every weather image through a multiclass classifier.



**Figure 2.** Architectural diagram of the proposed framework.

#### 3.1. Data Preparation Subsystem

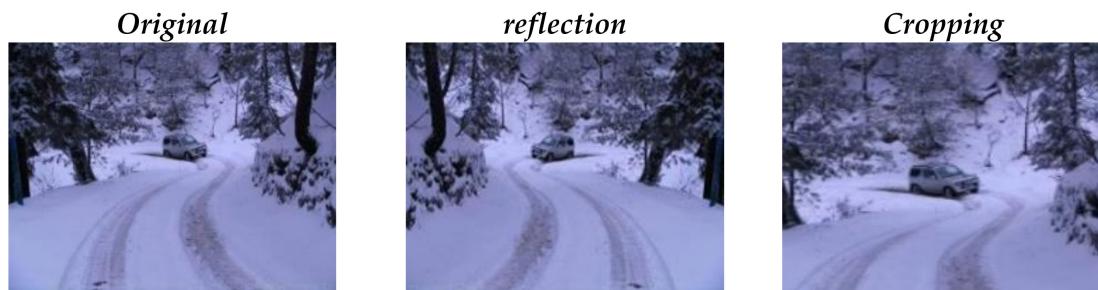
In this research, we have combined two main and common weather conditions datasets, DAWM2020 dataset [34] and MCWCD2018 [35], to end up with a dataset composed of 1656 image samples that are grouped into six classes for weather conditions: cloudy (300 images), rainy (215 images), snowy (204 images), sandy (319 images), shine (253 images), and sunrise (365 images). Figure 3 provides sample images for the different weather conditions in the dataset.



**Figure 3.** Sample images of different classes used in this work.

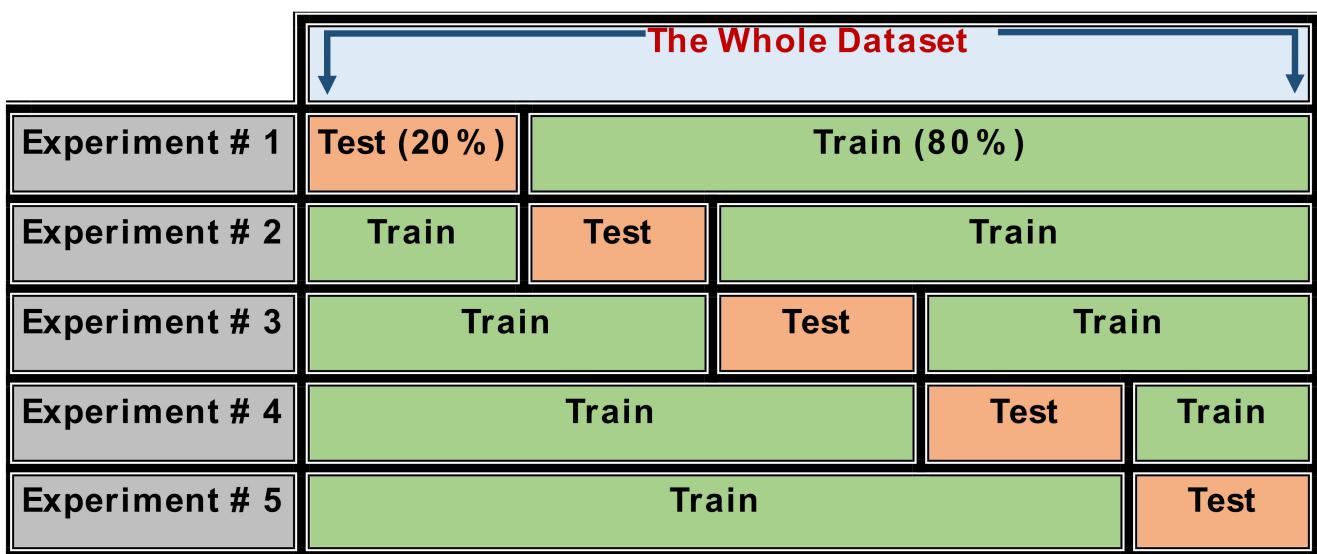
Once the image datasets are collected, combined, and labeled into six class labels by means of six folders, each of which holds the name of one weather class, the data are imported into MATLAB2021b and hosted within an image data store (IMD) that is local to the MATLAB platform. Thereafter, the collected colored images have undergone a series of consecutive preprocessing operations. Initially, the image-type of all images is unified to JPG image extension, and then the image-resize operation is applied over all JPG images in which the sizes of all images are converted to 3D matrices (RGB images) with image dimensions of  $224 \times 224 \times 3$ . The image resize is essential in order to accommodate the input image layer for the three utilized deep convolutional neural networks (SqueezeNet, ResNet-50, and EfficientNet-b0).

Now, as all images have the same type and size, data augmentation can be performed to effectively increase the amount of training data by applying randomized augmentation operations on the dataset. The augmentation process configures a set of preprocessing options such as resizing, cropping, rotation, reflection, invariant distortions, and others. Data augmentation is valuable to enhance the performance and outcomes of deep learning models by producing new and various examples to train datasets. If the dataset in a deep learning model is rich and sufficient, the model performs better and is more accurate [36]. Figure 4 illustrates some new images resulting from applying data augmentation preprocessing operations.



**Figure 4.** Sample images resulted in form data augmentation preprocessing.

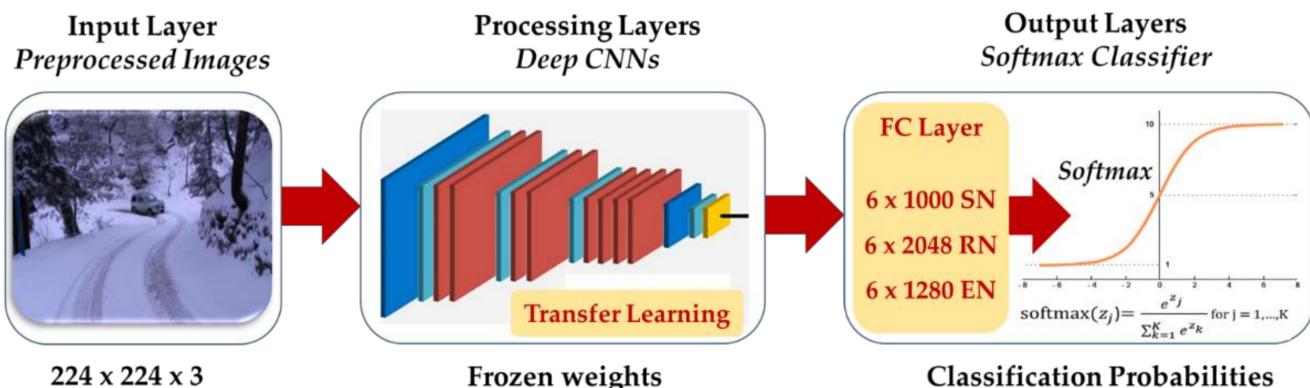
As for now, our IMD comprises all original images along with augmented images (generating almost an additional 5000 images), all of them are JPG unified images with the size of  $224 \times 224 \times 3$ , and the images are shuffled randomly. Image shuffling aims to mix up data to break the possible biases during data preparation (e.g., putting all the cloudy images first and then the rainy ones). Finally, the data are fed to the learning models into two image datasets: a training dataset that includes 80% of combined dataset images and is used to train the intelligent model and a testing dataset that includes 20% of combined dataset images and is used to test the model effectiveness prior to real-time deployment. However, to ensure a high level of assurance about the system performance, we have applied 5-fold cross-validations [37]. Here, the dataset is split into 5 folds. In the first iteration, the first fold is used to test the model and the rest are used to train the model. In the second iteration, 2nd fold is used as the testing set while the rest serve as the training set. Figure 5 demonstrates our 5-fold cross-validation policy.



**Figure 5.** The 5-fold cross validation process used in this research.

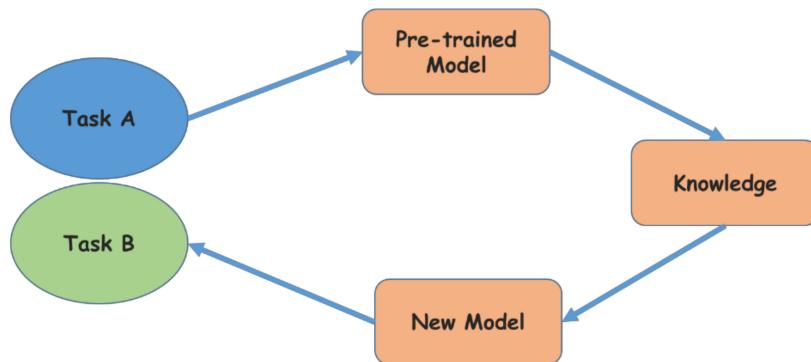
### 3.2. Learning Model Subsystem

This subsystem is concerned with training and validating the model to recognize the classes of weather conditions using deep supervised convolutional neural networks. Once the images are collected, combined, labeled, and preprocessed, they can be processed through the feedforward layers of the deep CNNs to produce the output classes. Figure 6 illustrates the main development stages of the learning subsystem.



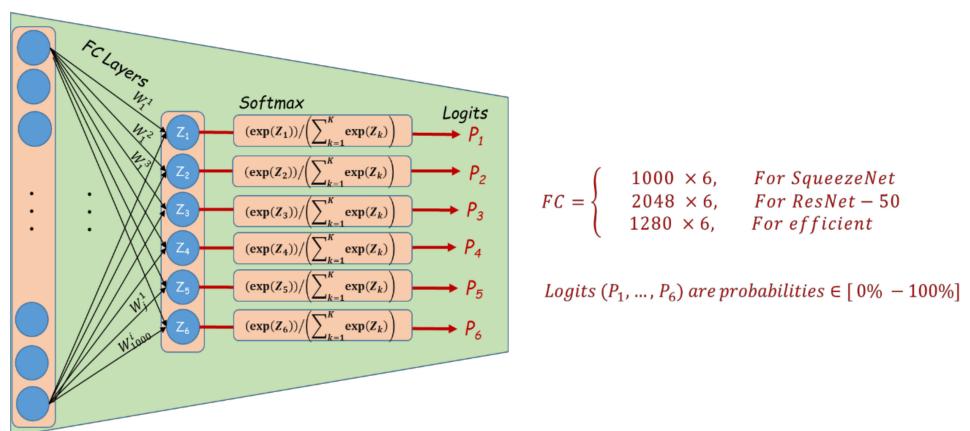
**Figure 6.** Development stages of the learning model subsystem.

- Input Layer: This layer is the first layer in the learning model subsystem. It represents the output of the data preparation subsystem that results in preprocessed RGB images of dimension  $224 \times 224 \times 3$  that accommodate the input of the employed CNNs.
- Processing Layer: This layer is concerned with feature extraction and learning to perform image classification for weather conditions. For optimized performance and to save time, we make use of transfer learning technology which provides a shortcut to saving time or obtaining better performance [38]. With transfer learning, there is no need to train the whole model from scratch for the new classification task; alternatively, it is conceivable to use any of the frozen weights resultant from pretraining the common deep neural networks on the ImageNet dataset [39]. However, fine-tuning for the learning parameters and the input and output layers is still needed for every specific application. Specifically, we transfer the learning of three deep CNNs: SqueezeNet CNN [40], ResNet-50 CNN [41], and EfficientNet-b0 CNN [42]. The main idea of transfer learning is demonstrated in Figure 7 below. The model is fully trained to do classification task A. The knowledge (pretrained parameters) is stored and transferred to the new model to do classification task B with fine-tuning.



**Figure 7.** Illustration of the main idea of transfer learning technique.

- Output Layer: This layer is concerned with trainable parameters (weights) of the pretrained deep CNNs. It should be compatible with both the number of output connections coming from every pretrained CNN and with the number of required predictable outputs. Figure 8 illustrates the connections at the output layer for every pretrained CNN. The output of SqueezeNet (1000) is fully connected with the number of classes (6), the output of ResNet-50 (2048) is fully connected with the number of classes (6), and the output of EfficientNet-B0 (1280) is fully connected with the number of classes (6). The final output will be provided as a SoftMax probability function, and the maximum probability will be selected to represent the final classification result.



**Figure 8.** Development of output layers in the learning model subsystem.

### 3.3. Evaluation and Deployment Subsystem

To measure the effectiveness of the proposed DL models, we used the standard evaluation metrics to evaluate the performance of every deep learning model in order to characterize its performance and to pick up the optimal model that provides the best evaluation metrics for deployment in real-time autonomous vehicles. The standard metrics are those calculated based on the confusion matrix analysis at the testing phases. Once the optimal model is deployed, it can be easily used to select the final output class from the results generated by the SoftMax probabilities. For example, if the SoftMax for a specific sample image produces the following results (Table 1), then the weather is classified to be rainy, and thus the autonomous vehicle can configure itself for working in a rainy environment.

**Table 1.** Sample output of Softmax classifier.

P <sub>Cloudy</sub>	P <sub>Rainy</sub>	P <sub>Snowy</sub>	P <sub>Sandy</sub>	P <sub>Sunrise</sub>	P <sub>Shine</sub>
1.25%	93%	1.75%	1.0%	1.2%	1.8%

In addition, we summarize the evaluation metrics used in this research in Figure 9. This includes the analysis of confusion matrix that correlates the relationship of true classes and predicted classes comprising the number of true predictions (TP, TN) and the number of false predictions (FN: Type I Error, FP: Type II Error). Accordingly, we derive the other metrics, such as model accuracy, model sensitivity (recall), model precision, and harmonic mean (F1-score) of the model, from the confusion matrix.

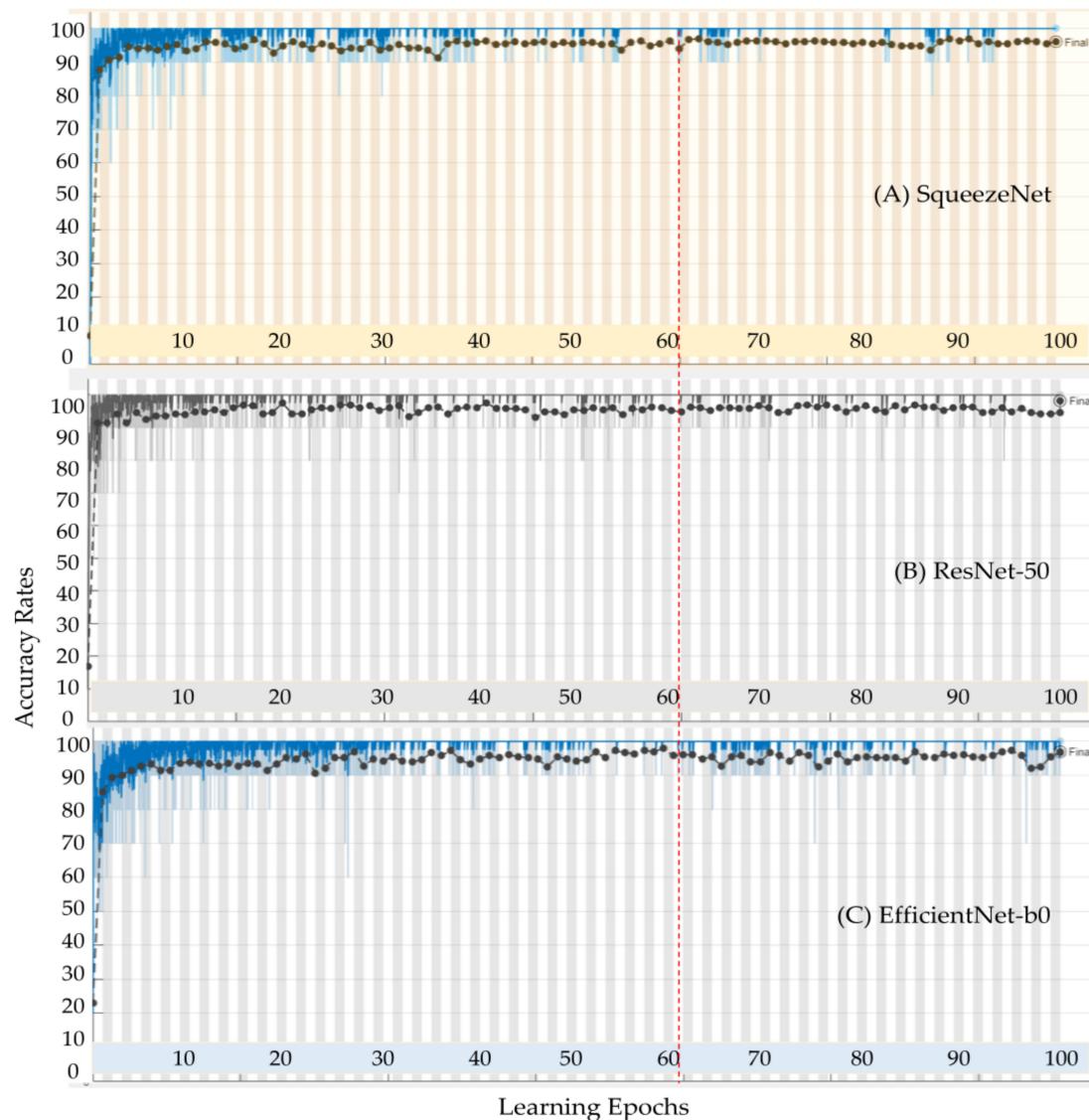
		Predicted Classes		$\frac{TP}{(TP + FN)}$
		Positive	Negative	
Actual Classes	Positive	True Positive (TP)	False Positive (FP)	$\frac{TP}{(TP + FN)}$
	Negative	False Negative (FN)	True Negative (TN)	$\frac{TN}{(TN + FP)}$
		$\frac{TP}{(TP + FP)}$	$\frac{2 \times (Precision \times Sensitivity)}{(Precision + Sensitivity)}$	$\frac{TP + TN}{(TP + FP + TN + FN)}$

**Figure 9.** Summary of evaluation metrics employed in this research.

## 4. Results and Discussion

In this work, we developed a computational intelligence model for detecting adverse weather conditions for autonomous vehicles via deep learning. To measure the effectiveness of the proposed approach, we have evaluated the system performance in terms of the evaluation metrics mentioned above, and we report our empirical results in this section. Initially, Figure 10 depicts the performance trajectories for the classification process of the proposed weather detection system using the (A) SqueezeNet CNN, (B) ResNet-50 CNN, and (C) EfficientNet-b0 CNN. Every subfigure shows the fulfillment plots for the learning process during the training and validation activities regarding detection accuracy and detecting loss using 100 learning epochs. According to the figure, all performance trajectories (i.e., accuracy curves) for all CNNs are systematically advancing along with the succeeding of learning epochs with an increasing inclination for the accuracy curves toward 100%. It can also be observed that all performance curves saturate after almost 60 learning process epochs, scoring accuracy rates of 100% and loss values of 0.0% for the training

activity of all CNN models as well as scoring accuracy rates of 98.48%, 97.78%, and 96.05% with corresponding loss (mean square error) values of 1.52%, 2.22%, and 3.95% for the SqueezeNet, ResNet-50, and EfficientNet-b0 models, respectively. These minor differences in saturation levels and thresholds for training accuracy/loss and testing accuracy/loss are permissible to preclude the model underfitting or overfitting [43].



**Figure 10.** Performance trajectories for models based on (A) SqueezeNet CNN, (B) ResNet-50 CNN, and (C) EfficientNet-b0 CNN.

In addition, Figure 11 demonstrates the six-class confusion matrix analysis for the validation/testing dataset for all developed models. The robustness of all models can be viewed through the large number of correctly predicted samples represented in the diagonal of the matrix (i.e., the number of true positives + the number of true negatives) compared to the small number of incorrectly predicted samples represented in the upper and lower areas of the diagonal (i.e., the number of false positives/Type II error + the number of false negatives/Type I error). For instance, the confusion matrix for the SqueezeNet-based model reported a total of 317 samples for the number of true positives/negatives and a total of 13 for the number of false positives/negatives out of 330, while the confusion matrix for the ResNet-50-based model reported a total of 326 samples for the number of true positives/negatives and a total of four for the number of false positives/negatives

out of 330. On the other hand, the confusion matrix for the EfficientNet-b0-based model reported a total of 323 samples for the number of true positives/negatives and a total of seven for the number of false positives/negatives out of 330. The obtained numbers confirm the effectiveness of the three models with a slight advantage for the model based on the ResNet-50 CNN.

True Classes	Predicted Classes					
	Cloudy	Rainy	Snowy	Sandy	Shine	Sunrise
Cloudy	62	1	0	1	0	0
Rainy	2	39	0	0	0	0
Snowy	0	0	60	1	0	0
Sandy	0	2	2	39	0	0
Shine	0	0	3	0	46	1
Sunrise	0	0	0	0	0	71

True Classes	Predicted Classes					
	Cloudy	Rainy	Snowy	Sandy	Shine	Sunrise
Cloudy	62	1	0	1	0	0
Rainy	0	41	0	0	0	0
Snowy	0	0	61	0	0	0
Sandy	0	1	0	42	0	0
Shine	0	0	1	0	49	0
Sunrise	0	0	1	0	0	70

True Classes	Predicted Classes					
	Cloudy	Rainy	Snowy	Sandy	Shine	Sunrise
Cloudy	64	0	0	0	0	0
Rainy	0	41	0	0	0	0
Snowy	2	0	58	1	0	0
Sandy	0	0	1	41	1	0
Shine	1	0	1	0	48	0
Sunrise	0	0	0	0	0	71

**Figure 11.** Six-class confusion matrix analysis for models based on (A) SqueezeNet CNN, (B) ResNet-50 CNN, and (C) EfficientNet-b0 CNN.

Consequently, based on the outcomes reported in the confusion matrix and according to the formulas stated in Figure 9, we calculated the other standard performance indicators such as detection accuracy, detection sensitivity, detection precision, and detection harmonic average (F1-score). Therefore, Table 2 summarizes and compares the obtained performance indication results for the proposed framework of 6-class weather detection using transfer-learning-based SqueezeNet CNN, ResNet-50 CNN, and EfficientNet-b0 CNN. The comparison exposes the dominance of the ResNet-50 CNN model in classifying weather conditions, scoring the highest performance results with 0.72–2.45%, 0.77–3.00%, 0.45–2.45%, and 0.60–2.76% improvement over other DL models in terms of accuracy, precision, sensitivity, and F1-score, respectively. Therefore, this model can be selected for deployment with autonomous vehicles to provide real-time weather detection to help the autonomous vehicles select their best operational model/configurations in response to weather conditions.

**Table 2.** Summary of key performance indicators (KPIs) for the proposed framework.

Model	Accuracy	Precision	Sensitivity	F1-Score
SqueezeNet	96.05%	95.51%	95.96%	95.68%
ResNet-50	98.48%	98.51%	98.41%	98.44%
EfficientNet-b0	97.78%	97.74%	97.96%	97.84%

Conclusively, even supposing the existing weather classification models reported in the literature employ different learning models, learning configurations, network architectures, learning policies, and development techniques, nevertheless, the comparison between the detection models can still be effective if we contrast them against common vital metrics such as detection accuracy and number of classes. Consequently, we provide summarized contrasting results of our best reported results (the ResNet-50 model) with the results reported by other related state-of-art models in Table 3. To date, we have limited our comparison with models developed mainly in the last 5 years. Based on the comparison stated in the table, it can be clearly observed that our ResNet-50-based weather detection model is remarkable, scoring the highest accuracy for the six-class weather detection model and improving the classification accuracy for the six weather conditions classifiers by a factor of 0.5–21%. In addition to this, a short detection time has been noted for the weather-detection-based ResNet-50 CNN model, involving an average of 5 (ms) for the

time-per-inference step using the GPU component. Consequently, the proposed framework can be effectively implemented in real-time environments to provide decisions in demand for autonomous vehicles with quick-precise detection capacity.

**Table 3.** Comparing our best results with other existing models in the literature.

Reference Number	Model Year	Classification Model	Intersection with the Used Dataset	Number of Classes	Accuracy Rates
[44]	2016	Discriminative dictionary learning	Partial	3 classes (sunny, cloudy, overcast)	94.00%
[27]	2017	Random forest classifiers	Total	4 classes (sunny, snowy, and rainy, cloudy)	96.30%
[45]	2017	GoogLeNet CNN	Partial	4 classes (sunny, blizzard, rainy, and foggy)	95.46%
[46]	2018	VGG-16 CNN	Total	4 classes (sunny, snowy, rainy, and cloudy)	94.71%
[32]	2018	GoogLeNet and AlexNet CNNs	Partial	3 classes (hazy, rainy, and snowy)	92.00%
[47]	2019	Stacked ensemble Algorithms	Total	4 classes (cloudy, rainy, shine, and sunrise)	86.00%
[29]	2019	ResNet-50 CNN	Partial	3 classes (clear, foggy, and snowy)	97.69%
[48]	2020	ResNet-101, and DenseNet121	Partial	9 classes (rain, dew, snow, frost, fog, ice, rain, fog, and hail)	81.25%
[28]	2020	ResNet-15 CNN	Partial	4 classes (sunny, snowy, rainy, and foggy)	96.03%
[49]	2020	ResNet-18 CNN	Total	4 classes (sunny, cloudy, shine, and sunrise)	98.20%
[30]	2021	Deep MeteCNN	Total + Additional classes	11 classes (rain, dew, snow, frost, fog, ice, rain, fog, hail, shine, and sunrise)	92.00%
Proposed	2022	ResNet-50 CNN	Combined Dataset	6 classes (cloudy, rainy, snowy, sandy, shine, and sunrise)	98.50%,

## 5. Conclusions and Future Directions

An intelligent independent deep-learning-based weather detection system to support the decision-making of autonomous vehicles system has been proposed, developed, and evaluated in this paper. The proposed system has characterized the performance capacities for three deep convolutional neural networks (CNNs), the SqueezeNet CNN, the ResNet-50 CNN, and the EfficientNet-b0 CNN. The models have been evaluated on a combined dataset from MCWDS2018 and DAWN2020 to provide six classes of the weather classification system, cloudy, rainy, snowy, sandy, shine, and sunrise. Extensive experiments have been conducted to investigate the quality of proposed models, and the experimental results demonstrated that all models are remarkable, providing accuracy rates of 98.48%, 97.78%, and 96.05% for the models based on ResNet-50, EfficientNet-b0, SqueezeNet, respectively. In addition, the empirical results revealed that our model is a high-performing model, providing elevated performance factors with a short time-per-inference step using the GPU component. In this research, we have combined two datasets, namely, the DAWN2020 dataset and MCWCD2018, to produce six classes. For the DAWN2020 dataset, to the best of our knowledge, this is the first time this dataset has been used in deep learning applications, while MCWCD2018 has been used several times to produce four classes. Therefore, the majority of the compared models in our table either use the four-class MCWCD2018 dataset that we employed (total intersection), use a part of it, or have it combined with some other datasets (partial intersection). Eventually, our main objective is to have an accurate model with minimum inferencing delay to detect weather conditions and help autonomous systems in this portion of the design. Though there is no standard dataset for weather

conditions to follow in all detection systems, the accuracy metric is vital for comparing the performance of several models in this regard. In the future, we intend to test our lightweight architecture on a variety of devices to evaluate the model in terms of processing time and model accuracy. Additionally, the suggested system can be developed to detect a variety of objects on the street, such as signs, mailboxes, and street lighting, by extending our experiments on other autonomous driving datasets in challenging scenarios with miscellaneous objects on the street to improve the system's usability in the real world.

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

**Funding:** This research received no external funding.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

## References

- Wang, H.; Yu, Y.; Cai, Y.; Chen, X.; Chen, L.; Liu, Q. A Comparative Study of State-of-the-Art Deep Learning Algorithms for Vehicle Detection. *IEEE Intell. Transp. Syst. Mag.* **2019**, *11*, 82–95. [[CrossRef](#)]
- Hossain, S.; Lee, D.-J. Deep Learning-Based Real-Time Multiple-Object Detection and Tracking from Aerial Imagery via a Flying Robot with GPU-Based Embedded Devices. *Sensors* **2019**, *19*, 3371. [[CrossRef](#)] [[PubMed](#)]
- Abu Al-Haija, Q.; Samad, M.D. Efficient LuxMeter Design Using TM4C123 Microcontroller with Motion Detection Application. In Proceedings of the 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 7–9 April 2020; pp. 331–336. [[CrossRef](#)]
- Hassaballah, M.; Kenk, M.A.; Muhammad, K.; Minaee, S. Vehicle Detection and Tracking in Adverse Weather Using a Deep Learning Framework. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 4230–4242. [[CrossRef](#)]
- Kenk, M.A.; Hassaballah, M. DAWN: Vehicle Detection in Adverse Weather Nature Dataset. *arXiv* **2020**, arXiv:2008.05402.
- McVeigh, J.; Bates, G.; Chandler, M. *Steroids and Image Enhancing Drugs*; Centre for Public Health, Liverpool John Moores University: Liverpool, UK, 2015.
- Atkinson, A.M.; van de Ven, K.; Cunningham, M.; de Zeeuw, T.; Hibbert, E.; Forlini, C.; Barkoukis, V.; Sumnall, H. Performance and image enhancing drug interventions aimed at increasing knowledge among healthcare professionals (HCP): Reflections on the implementation of the Dopinglinkki e-module in Europe and Australia in the HCP workforce. *Int. J. Drug Policy* **2021**, *95*, 103141. [[CrossRef](#)] [[PubMed](#)]
- Sindhu, V.S. Vehicle Identification from Traffic Video Surveillance Using YOLOv4. In Proceeding of the 5th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 6–8 May 2021.
- Wang, D.; Wang, J.-G.; Xu, K. Deep Learning for Object Detection, Classification and Tracking in Industry Applications. *Sensors* **2021**, *21*, 7349. [[CrossRef](#)]
- Blettery, E.; Fernandes, N.; Gouet-Brunet, V. How to Spatialize Geographical Iconographic Heritage. In Proceedings of the 3rd Workshop on Structuring and Understanding of Multimedia heritAge Contents, Online, 17 October 2021.
- Lou, G.; Deng, Y.; Zheng, X.; Zhang, T.; Zhang, M. An investigation into the state-of-the-practice autonomous driving testing. *arXiv* **2021**, arXiv:2106.12233.
- Fan, R.; Djuric, N.; Yu, F.; McAllister, R.; Pitas, I. Autonomous Vehicle Vision 2021: ICCV Workshop Summary. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
- Lei, D.; Kim, S.-H. Application of Wireless Virtual Reality Perception and Simulation Technology in Film and Television Animation. *J. Sensors* **2021**, *2021*, 1–12. [[CrossRef](#)]
- Lee, L.-H.; Braud, T.; Zhou, P.; Wang, L.; Xu, D.; Lin, Z.; Kumar, A.; Bermejo, C.; Hui, P. All one needs to know about metaverse: A complete survey on technological singularity, virtual ecosystem, and research agenda. *arXiv* **2021**, arXiv:2110.05352.
- Cohen, S.; Lin, Z.; Ling, M. Utilizing Object Attribute Detection Models to Automatically Select Instances of Detected Objects in Images. U.S. Patent No. 11,107,219, 31 August 2021.
- Chang, W.W.T.; Cohen, S.; Lin, Z.; Ling, M. Utilizing Natural Language Processing and Multiple Object Detection Models to Automatically Select Objects in Images. U.S. Patent Application No. 16/800,415, 26 August 2021.
- Villa, F.; Severini, F.; Madonini, F.; Zappa, F. SPADs and SiPMs Arrays for Long-Range High-Speed Light Detection and Ranging (LiDAR). *Sensors* **2021**, *21*, 3839. [[CrossRef](#)]

18. Abu Al-Haija, Q.; Al-Saraireh, J. Asymmetric Identification Model for Human-Robot Contacts via Supervised Learning. *Symmetry* **2022**, *14*, 591. [[CrossRef](#)]
19. Bayoudh, K.; Knani, R.; Hamdaoui, F.; Mtibaa, A. A survey on deep multimodal learning for computer vision: Advances, trends, applications, and datasets. *Vis. Comput.* **2021**, *1*–32. [[CrossRef](#)] [[PubMed](#)]
20. Jiang, H.; Hu, J.; Liu, D.; Xiong, J.; Cai, M. DriverSonar: Fine-Grained Dangerous Driving Detection Using Active Sonar. In Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, Online, 14 September 2021; Volume 5, pp. 1–22.
21. Gupta, A.; Anpalagan, A.; Guan, L.; Khwaja, A.S. Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues. *Array* **2021**, *10*, 100057. [[CrossRef](#)]
22. Sharma, T.; Debaque, B.; Duclos, N.; Chehri, A.; Kinder, B.; Fortier, P. Deep Learning-Based Object Detection and Scene Perception under Bad Weather Conditions. *Electronics* **2022**, *11*, 563. [[CrossRef](#)]
23. Abu Al-Haija, Q.; Krichen, M.; Abu Elhaija, W. Machine-Learning-Based Darknet Traffic Detection System for IoT Applications. *Electronics* **2022**, *11*, 556. [[CrossRef](#)]
24. Garg, M.; Ubhi, J.S.; Aggarwal, A.K. Deep Learning for Obstacle Avoidance in Autonomous Driving. In *Autonomous Driving and Advanced Driver-Assistance Systems (ADAS)*; CRC Press: Boca Raton, FL, USA, 2021; pp. 233–246.
25. Abu Al-Haija, Q.; Al-Dala’ien, M. ELBA-IoT: An Ensemble Learning Model for Botnet Attack Detection in IoT Networks. *J. Sens. Actuator Netw.* **2022**, *11*, 18. [[CrossRef](#)]
26. Elhoseiny, M.; Huang, S.; Elgammal, A. Weather classification with deep convolutional neural networks. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 3349–3353. [[CrossRef](#)]
27. Chu, W.-T.; Zheng, X.-Y.; Ding, D.-S. Camera as weather sensor: Estimating weather information from single images. *J. Vis. Commun. Image Represent.* **2017**, *46*, 233–249. [[CrossRef](#)]
28. Xia, J.; Xuan, D.; Tan, L.; Xing, L. ResNet15: Weather Recognition on Traffic Road with Deep Convolutional Neural Network. *Adv. Meteorol.* **2020**, *2020*, 1–11. [[CrossRef](#)]
29. Ibrahim, M.R.; Haworth, J.; Cheng, T. WeatherNet: Recognising Weather and Visual Conditions from Street-Level Images Using Deep Residual Learning. *ISPRS Int. J. Geo-Inform.* **2019**, *8*, 549. [[CrossRef](#)]
30. Xiao, H.; Zhang, F.; Shen, Z.; Wu, K.; Zhang, J. Classification of Weather Phenomenon from Images by Using Deep Convolutional Neural Network. *Earth Space Sci.* **2021**, *8*, e2020EA001604. [[CrossRef](#)]
31. Roser, M.; Moosmann, F. Classification of weather situations on single color images. In Proceedings of the IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, 4–6 June 2008; pp. 798–803. [[CrossRef](#)]
32. Kang, L.-W.; Chou, K.-L.; Fu, R.-H. Deep Learning-Based Weather Image Recognition. In Proceedings of the 2018 International Symposium on Computer, Consumer and Control (IS3C), Taichung, Taiwan, 6–8 December 2018; pp. 384–387. [[CrossRef](#)]
33. Guerra, J.C.V.; Khanam, Z.; Ehsan, S.; Stolkin, R.; McDonald-Maier, K. Weather Classification: A new multi-class dataset, data augmentation approach and comprehensive evaluations of Convolutional Neural Networks. In Proceedings of the NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Edinburgh, UK, 6–9 August 2018; pp. 305–310. [[CrossRef](#)]
34. Gbeminiyi, A. MCWCD: Multi-Class Weather Conditions Dataset for Image Classification. *Mendeley Data*. 2018. Available online: <https://data.mendeley.com/datasets/4drtyfjtfy/1> (accessed on 17 January 2022).
35. Kenk, M. DAWN: Detection in Adverse Weather Nature. *Mendeley Data*. 2020. Available online: <https://data.mendeley.com/datasets/766ygrbt8y/3> (accessed on 17 January 2022).
36. Takimoglu, A. What is Data Augmentation? *Techniques & Examples in 2022, AI Multiple: Data-driven, Transparent, Practical New Tech Industry Analysis*. 2022. Available online: <https://research.aimultiple.com/data-augmentation/> (accessed on 17 January 2022).
37. Abu Al-Haija, Q.; Smadi, A.A.; Allehyani, M.F. Meticulously Intelligent Identification System for Smart Grid Network Stability to Optimize Risk Management. *Energies* **2021**, *14*, 6935. [[CrossRef](#)]
38. Al-Haija, Q.A.; Smadi, M.; Al-Bataineh, O.M. Identifying Phasic dopamine releases using DarkNet-19 Convolutional Neural Network. In Proceedings of the 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Toronto, ON, Canada, 21–24 April 2021; pp. 1–5. [[CrossRef](#)]
39. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [[CrossRef](#)]
40. Iandola, N.F.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
42. Tan, M.; Quoc, L. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
43. Abu Al-Haija, Q.; Adebanjo, A. Breast Cancer Diagnosis in Histopathological Images Using ResNet-50 Convolutional Neural Network. In Proceedings of the IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Vancouver, BC, Canada, 9–12 September 2020; pp. 1–7. [[CrossRef](#)]

44. Zheng, C.; Zhang, F.; Hou, H.; Bi, C.; Zhang, M.; Zhang, B. Active Discriminative Dictionary Learning for Weather Recognition. *Math. Probl. Eng.* **2016**, *2016*, 1–12. [[CrossRef](#)]
45. Zhu, Z.; Li, J.; Zhuo, L.; Zhang, J. Extreme Weather Recognition Using a Novel Fine-Tuning Strategy and Optimized GoogLeNet. In Proceedings of the 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Sydney, NSW, Australia, 29 November–1 December 2017; pp. 1–7.
46. Shi, Y.; Li, Y.; Liu, J.; Liu, X.; Murphrey, Y.L. Weather Recognition Based on Edge Deterioration and Convolutional Neural Networks. In Proceedings of the 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 2438–2443. [[CrossRef](#)]
47. Oluwafemi, A.G.; Zenghui, W. Multi-Class Weather Classification from Still Image Using Said Ensemble Method. In Proceedings of the Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA), Bloemfontein, South Africa, 28–30 January 2019; pp. 135–140. [[CrossRef](#)]
48. Wang, Y.; Li, Y. Research on Multi-class Weather Classification Algorithm Based on Multi-model Fusion. In Proceedings of the IEEE 4th Information Technology Networking Electronic and Automation Control Conference (ITNEC), Chongqing, China, 12–14 June 2020; pp. 2251–2255.
49. Abu Al-Haija, Q.; Smadi, M.A.; Zein-Sabatto, S. Multi-Class Weather Classification Using ResNet-18 CNN for Autonomous IoT and CPS Applications. In Proceedings of the International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 16–18 December 2020; pp. 1586–1591. [[CrossRef](#)]