sustainability

MDPI

Article

Convolutional Neural Network-Based Vehicle Classification in Low-Quality Imaging Conditions for Internet of Things Devices

Bamoye Maiga [1], Yaser Dalveren [2], Ali Kara [3] and Mohammad Derawi [4,*]

[1] Graduate School of Natural and Applied Sciences, Department of Electrical and Electronics Engineering, Atilim University, Ankara 06830, Turkey; bamoyemeiga@gmail.com
[2] Department of Electrical and Electronics Engineering, Atilim University, Ankara 06830, Turkey; yaser.dalveren@atilim.edu.tr
[3] Department of Electrical and Electronics Engineering, Gazi University, Ankara 06570, Turkey; akara@gazi.edu.tr
[4] Department of Electronic Systems, Norwegian University of Science and Technology, 2815 Gjovik, Norway
* Correspondence: mohammad.derawi@ntnu.no

Abstract: Vehicle classification has an important role in the efficient implementation of Internet of Things (IoT)-based intelligent transportation system (ITS) applications. Nowadays, because of their higher performance, convolutional neural networks (CNNs) are mostly used for vehicle classification. However, the computational complexity of CNNs and high-resolution data provided by high-quality monitoring cameras can pose significant challenges due to limited IoT device resources. In order to address this issue, this study aims to propose a simple CNN-based model for vehicle classification in low-quality images collected by a standard security camera positioned far from a traffic scene under low lighting and different weather conditions. For this purpose, firstly, a new dataset that contains 4800 low-quality vehicle images with 100 × 100 pixels and a 96 dpi resolution was created. Then, the proposed model and several well-known CNN-based models were tested on the created dataset. The results demonstrate that the proposed model achieved 95.8% accuracy, outperforming Inception v3, Inception-ResNet v2, Xception, and VGG19. While DenseNet121 and ResNet50 achieved better accuracy, their complexity in terms of higher trainable parameters, layers, and training times might be a significant concern in practice. In this context, the results suggest that the proposed model could be a feasible option for IoT devices used in ITS applications due to its simple architecture.

Keywords: bad weather; deep learning; intelligent transportation system; tiny images

check for updates

Citation: Maiga, B.; Dalveren, Y.; Kara, A.; Derawi, M. Convolutional Neural Network-Based Vehicle Classification in Low-Quality Imaging Conditions for Internet of Things Devices. *Sustainability* **2023**, *15*, 16292. https://doi.org/10.3390/su152316292

Academic Editor: Jiageng Ruan

Received: 26 October 2023
Revised: 14 November 2023
Accepted: 24 November 2023
Published: 24 November 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

The use of intelligent transportation system (ITS) applications, such as traffic flow monitoring [1,2], automated parking systems [3,4], and autonomous driving [5,6], has gained increasing interest over the last few years. Obviously, vehicle classification has an important role in the efficient implementation of ITS applications [7–9]. For this reason, various vehicle classification methods have been proposed in the literature. Vehicle classification methods are mainly categorized as vision-based, sound-based, remote sensing, contact-based, off-road-based, and hybrid methods [10]. Among these methods, vision-based methods are mostly studied in the literature due to their lower system installation, lower maintenance and operational costs, and higher efficiency [11].

In vision-based vehicle classification methods, image sequences of traffic scenes recorded by various types of cameras, which are able to feature the visual characteristics of a vehicle, are used. Typically, a vision-based method comprises two main stages. In the first stage, visual features are obtained by utilizing hand-crafted extraction methods. In the second stage, classification is performed by classical machine learning classifiers trained on the extracted features. Although vision-based methods perform well when compared to other existing classification methods, larger datasets and prior knowledge are strictly

required in order to maintain high performance and consistency. Therefore, deep learning (DL)-based methods, which are able to directly learn feature characteristics from large-scale image datasets, have been introduced to achieve better classification accuracy [12].

Currently, the use of DL approaches based on convolutional neural networks (CNNs) is the current trend. Particularly, CNNs are widely used for vehicle image classification because of their ability to learn hierarchical features from images. In fact, the use of CNNs is very popular among the research community due to the several reasons, such as working well on both large and small datasets, extracting both low- and high-level features from images, and achieving the same performance with fewer computational resources compared to other DL approaches. They have been shown to achieve state-of-the-art performance on a variety of vehicle image datasets [13–24]. In [13], a method based on a semi-supervised CNN is provided for vehicle classification from vehicle frontal view images. In [14], a fine-grained vehicle classification model is proposed for ITS applications. A CNN-based vehicle classification method consisting of pre-training and fine-tuning steps is proposed in [15]. In [16], a deep CNN model is presented for vehicle classification in traffic surveillance systems. In [17], a CNN framework is proposed for vehicle type classification and vehicle color classification in surveillance videos. In [18], a Faster Region convolutional neural network (Faster R-CNN) is proposed for vehicle classification. Moreover, an improved Faster R-CNN method for vehicle classification is proposed in [19]. A modified R-CNN for vehicle classification is also proposed in [20]. To improve the performance of fine-grained vehicle classification, a channel max pooling approach is presented in [21]. In [22], the use of various CNN-based models on non-laned heterogeneous traffic images for vehicle classification is presented. Moreover, a vehicle classification model based on transfer learning with a deep CNN is proposed in [23].

On the other hand, the Internet of Things (IoT) has now become a very popular concept that has important potential to reconstruct various areas of daily life [25]. Nowadays, various IoT-based ITS applications, such as safer roads [26], autonomous vehicles [27], parking management [28], and traffic management [29], have been proposed in the literature. In IoT-based ITS applications, a network of a large number of interconnected electronic devices, including cameras, smartphones, or sensors, can be used. For this reason, massive amounts of data are expected to be generated. Then, it is essential to effectively process and analyze such large amounts of data. Here, a CNN-based framework (model) could be a good choice due to its higher performance on large-scale datasets. However, using a CNN-based model might be computationally complex in practice. This significantly prohibits its usage on edge devices in IoT-based ITS applications, where the limited computational capabilities and resources are the most significant concerns. Hence, it is strictly necessary to choose a simple CNN-based model for IoT-based ITS applications.

The methods presented in [13–24] are proposed for real-time ITS applications that use high-quality monitoring cameras. Although these types of cameras provide high-resolution (or frame rate) videos or images, their higher cost leads to an important concern in the deployment of IoT-based ITS applications. Instead, when low-cost surveillance cameras are used alternatively, significant challenges are expected to be faced in the classification process. Particularly, these challenges can be aroused by various factors, including the low quality of images due to image blurring, low or adverse lighting conditions, and various weather conditions, such as hazy, rainy, or snowy scenes. To overcome these challenges, a few CNN-based classification methods have been introduced in the literature [30–36]. Among these, only the study presented in [36] addresses vehicle classification for low-quality images collected by a low-cost and low-resolution surveillance camera used for security purposes rather than traffic monitoring. However, poor imaging conditions, such as low lighting conditions and various weather conditions (hazy, rainy, or snowy), are not taken into account in vehicle classification.

This article, as a follow-up study of [36], is devoted to proposing a simple but efficient CNN-based model for vehicle classification in surveillance images collected by a standard security camera positioned distant from the traffic scene under various imaging conditions.

The primary goal is to minimize the number of trainable parameters and reduce resource usage, which are the most critical concerns for the effective deployment of ITS on IoT devices that have limited computing power and memory resources. For this purpose, a new dataset, which contains 4800 low-quality vehicle images with 100 × 100 pixels and a 96 dpi resolution collected in low lighting (adverse illumination and nightlight) and different weather conditions (hazy and rainy), was created. Then, the proposed model and other well-known CNN-based models, such as DenseNet121, ResNet50, Inception v3, Inception-ResNet v2, Extreme Inception (Xception), and VGG19, were tested on the created dataset. Next, their performances were comparatively assessed in terms of accuracy and complexity metrics. The main contributions of the study presented in this article can be summarized as follows:

- A simple CNN-based model is proposed for the classification of low-quality vehicle images in low lighting and adverse weather conditions.
- A new dataset containing low-quality vehicle images collected in various environmental conditions is created.
- The efficiency of the proposed model is verified by conducting a comparative analysis with several well-known CNN-based models. Due to its lower computational requirements and acceptable accuracy, it is shown that the proposed model could be implemented on edge devices in IoT-based ITS applications.

The remainder of this paper is structured as follows: In Section 2, the related works presented in the literature are reviewed. In Section 3, the proposed model is presented. This is followed by Section 4, where the details of the experiments on the created dataset are described. Then, experimental results are discussed in Section 5. Next, in Section 6, discussions based on the achieved results are summarized. Lastly, concluding remarks are provided in Section 7.

2. Related Work and Problem Statement

As mentioned in the previous section, one of the most important difficulties for vision-based vehicle classification methods is the low resolution of the data due to environmental conditions such as low lighting and undesirable weather conditions. In this context, several CNN-based models have been proposed in the literature to solve this difficulty [30–36]. In [30], the performance of a typical CNN architecture in vehicle detection and classification using samples obtained from low-resolution traffic videos is investigated. The sample used in the experiments has a resolution of 704 × 480 pixels. In the experiments, various traffic and weather conditions are considered. In [31], a model based on the AdaBoost algorithm and deep CNNs, including VGGNet, AlexNet, and GoogLeNet, is proposed to classify vehicle images with a resolution of 224 × 224 pixels captured in various conditions, such as rain, haze, and night. In [32], a CNN model is proposed for vehicle classification with low-resolution images. The samples used in the experiments are obtained by decreasing the vision quality of high-resolution images of the BIT-vehicle dataset [13]. Particularly, all the samples selected from the dataset are resized to a resolution of 32 × 32 pixels. In [33], a method based on Faster R-CNN for vehicle detection and classification in real-time applications is proposed. The proposed method is tested on field videos (720 × 480 pixels, at 25 fps) under different weather conditions, such as rain, day, and night. In [34], another method that employs a Generative Adversarial Network (GAN) is proposed to classify tiny vehicles. The idea behind the method is to generate high-resolution images from distant vehicles, which are fuzzy and blurred because of their low resolutions. In [35], a CNN-based vehicle classification architecture is proposed for real-time ITS applications in adverse light conditions. Different types of data augmentation are applied to expand the dataset containing vehicle images with a resolution of 224 × 224 pixels. As a summary, Table 1 lists the relevant works and provides information for the utilized models, data properties, environmental conditions, and settings.

Table 1. Summary of relevant works.

Ref.	Model	Data/Image Properties	Environmental Conditions	Camera Settings
[30]	CNN	• 28 × 28 pixels • Random-view images	• Daylight • Sunny and rainy weather	Low-resolution cameras set close to the ROI with a depression angle view
[31]	AdaBoost algorithm and deep CNNs	• 224 × 224 pixels • Rear-view images	• Daylight and nightlight • Rainy and hazy weather	Traffic surveillance cameras set close to the ROI with a depression angle view
[32]	Modified CNN	• 32 × 32 pixels • Frontal-view images	• Daylight and nightlight • No weather conditions	
[33]	Faster R-CNN	• Multi-resolution images, including image widths of 1056, 864, 512, and 320 • Frontal- and random-view images	• Daylight and nightlight • Rainy weather	Traffic surveillance cameras set close to the ROI with both depression angle and dashcam view
[34]	GAN	• No resolution information • Random-view images	• Daylight and nightlight • No weather conditions	Cameras set close to the ROI with a dashcam view
[35]	ResNet	• 224 × 224 pixels • Random-view images	• Daylight • No weather conditions	Traffic surveillance cameras set close to the ROI with a depression angle view
[36]	Modified CNN	• 100 × 100 pixels • Random-view images	• Daylight • No weather conditions	Security surveillance camera distant from the ROI with a wide depression angle view
Our Work	Modified CNN		• Daylight and nightlight • Rainy and hazy weather	

As can be deduced from Table 1, in [30–35], the vehicle image data used for vehicle classification are collected by monitoring or surveillance cameras that are perfectly oriented toward the region of interest (ROI), such as a traffic scene or a road. It should be noted that the cameras used in data collection are not distant enough from the ROI. Moreover, the proposed models are relatively heavyweight, which might make them unsuitable to be implemented on edge devices in IoT-based ITS applications that have limited computational capabilities and resources. Therefore, in order to address these issues, the study presented in [36] proposes a simple CNN-based model for vehicle classification in low-resolution images collected by a standard surveillance camera installed far from the ROI. Particularly, it is shown for the first time in the literature that vehicle classification is possible with tiny and low-resolution images collected by a low-cost camera used for security purposes rather than traffic monitoring. However, one of the major challenges in vehicle classification, which can be attributed to the low resolution of images due to environmental conditions, such as low lighting and various weather conditions, is not addressed. Thus, it becomes very important to provide a simple CNN-based approach in order to overcome this challenge.

3. Overview of the Proposed CNN-Based Model

IoT-based ITS applications require large networks of interconnected devices to generate massive data. As mentioned earlier, CNN-based models are suitable for processing large datasets. However, their computational complexity limits their use on edge devices.

Therefore, simple CNN-based models might be preferred for IoT-based ITS applications due to the limited computational capabilities and resources.

On the other hand, in this study, the main intention is to make vehicle classification possible with tiny and low-resolution images collected by a low-cost camera, which are used for security purposes rather than traffic monitoring, in low lighting and various weather conditions. To this end, a CNN-based model that has a simple architecture is proposed. In this way, it is expected that the proposed model could provide useful insights to overcome the challenges associated with accuracy and speed requirements for IoT-based ITS applications.

Figure 1 shows the architecture of the proposed model. As can be seen from the figure, the architecture is mainly composed of two stages: (a) a feature extraction network, and (b) a regularization and output layer.

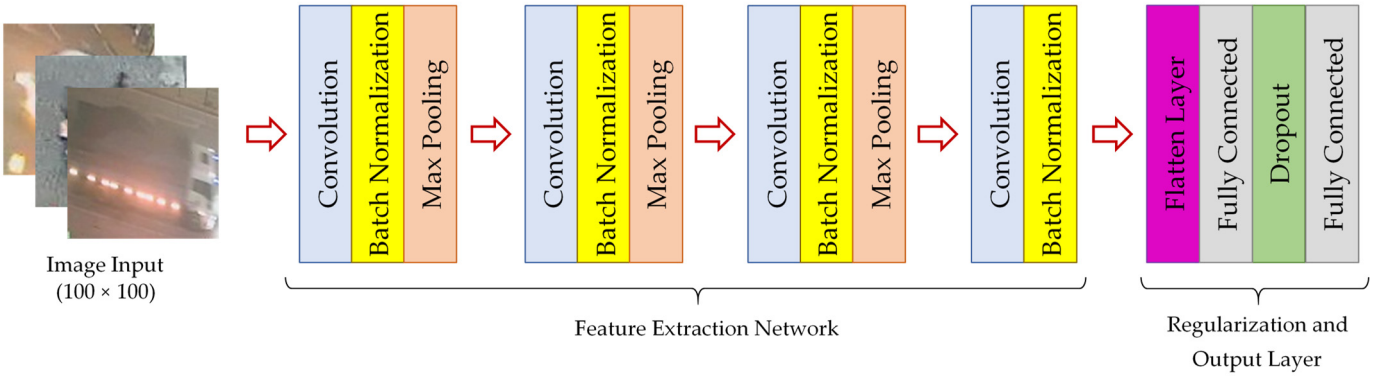


Figure 1. The architecture of the proposed model.

In the first stage of the architecture, four convolutional layers, four batch normalization layers, and three max pooling layers are used. This stage corresponds to the feature extraction network, where each convolutional layer has a 3×3 filter size. The convolutional layers used for feature extraction are designed in such a way that the depth increases with each additional layer. The first convolution layers (called shallow layers) extract macro details, while the last convolution layers (called deep layers) extract micro features. Such a design reduces the number of parameters by avoiding the transfer of unnecessary features to the next layer downstream while maintaining good performance. The first two convolution layers have a depth of 32 and 128, respectively, whereas the last two layers both have a depth of 512. Rectifier linear unit (ReLU) is used as the activation function for convolutional layers. Each convolutional layer uses a stride value of 1, and the padding is set to be the same in order to keep the output size of the convolutional layers the same as the input size. Batch normalization is used to stabilize the network. It allows the use of a much higher learning rate, which can speed up the training process. As shown in the figure, a batch normalization layer is used after each convolution layer. Moreover, in the network, down-sampling is performed by max pooling layers deployed after the first three convolutional layer groups. The first max pooling layer has a stride value of 4 and a pool size of 4, while the next two have a stride value of 2 and a filter size of 2×2.

In the second stage of the architecture, a flatten operation is applied after the convolutional and batch normalization layers to reshape the output of the first stage into a single-dimensional feature vector to be passed to a fully connected layer. The flatten operation is performed by the flatten layer from the TensorFlow and Keras libraries. This layer takes a three-dimensional matrix ($6 \times 6 \times 512$) inherited from the batch normalization layer that precedes it, and reduces it to a one-dimensional vector ($1 \times 18{,}432$). To prevent overfitting, the fully connected layer consisting of 16 hidden units is used, where ReLU is utilized as the activation function. A dropout layer is then used to randomly drop out hidden units. As is known, the dropout layer is mainly used to avoid overfitting. In the proposed model, the dropout layer is also used to act as a regularization method, approximating

the concurrent training of many nodes in the fully connected layer. During training, some neurons in the fully connected layer are ignored or dropped at random. For each epoch, the layer update during training is carried out with a different perspective. Therefore, the dropout layer is beneficial in order to break apart circumstances in which network tiers co-adapt to fix mistakes committed by prior layers, making the model more robust. Since the hidden fully connected layer has 16 units, while the output fully connected layer has five units, the dropout layer is implemented between these two layers. A high dropout rate would be too severe for such a network with a low number of units, while a very low dropout rate would not provide all the benefits of the dropout layer in order to avoid overfitting. Therefore, in the proposed model, the dropout rate is set to 0.3, which means that 30% of the inputs will be randomly dropped in each epoch. After the dropout layer, a final fully connected layer with six units (classes) is used, where the Softmax function is applied for classification.

4. Experiments

Experiments were conducted to evaluate the efficiency of the proposed model. In this section, the whole process followed in the experiments is described.

4.1. Dataset and Preprocessing

In order to test the proposed model, a new dataset containing vehicle images under low-quality imaging conditions was created [37]. Similar to [36], before creating the dataset, a set of video recordings was captured by a low-cost security surveillance camera monitoring a particular square in Konya, Turkey. The camera, which has a wide depression angle view, was mounted on one of the minarets of a mosque near the square. The installation height of the camera was 12 m. The position of the camera is shown in Figure 2a.

The video recordings were acquired in various conditions, such as daylight and nightlight, along with rain and haze. As an example, the views from the camera in daylight haze, daylight rain, and nightlight rain are shown in Figures 2b, 2c and 2d, respectively. It is clear from the figures that the camera placed on the minaret was distant from the traffic scene. Here, the traffic scene was considered the ROI for further processing.

The data acquisition was followed by cropping the vehicle images from the recordings. Specifically, the vehicle images with a 96 dpi resolution were manually cropped from each of the video frames. All the cropped images were then grouped into six classes, namely, bike, car, juggernaut, minibus, pickup, and truck. In this way, 800 vehicle images were collected for each class, resulting in a dataset containing 4800 vehicle images. In Figure 3, the samples of vehicles cropped from the video frames are shown.

Before feeding the data into the network during the training process, the data were preprocessed using a simple approach. The pre-processing step is essential to prepare the data before the training process. Thus, the images were divided into classes so that the model could learn to distinguish between different types of vehicles. The images were also normalized to ensure that they were all of the same size and format. In this context, each class was indexed to encode the data. After encoding, the data were resized to 100×100 pixels. The features and the corresponding labels were then separated from each other. Next, the features were normalized to ensure the stability of the training process.

4.2. Implementation Details

Before the implementation, the dataset was separated into training, validation, and test sets. It is important to note that the test set was used to determine whether the trained model could generalize its findings to other new data, whereas the validation set was used to tune the network hyperparameters, excluding parameters and learnable values (weights and biases). The training set consisted of 3360 vehicle images (70%) while the test set consisted of 960 vehicle images (20%) and the validation set consisted of the remaining 480 vehicle images (10%).

For the implementation, the Python programming language with the TensorFlow and Keras libraries in the VS Code platform was used. In the implementation, the workstation was equipped with an Intel® Core™ i5-9400F CPU with a 2.90 GHz processor, a NVIDIA Corporation TU116 (GeForce GTX 1660 Ti) GPU, 6 GB, and the Ubuntu 22.04.1 LTS (64-bit) operating system.

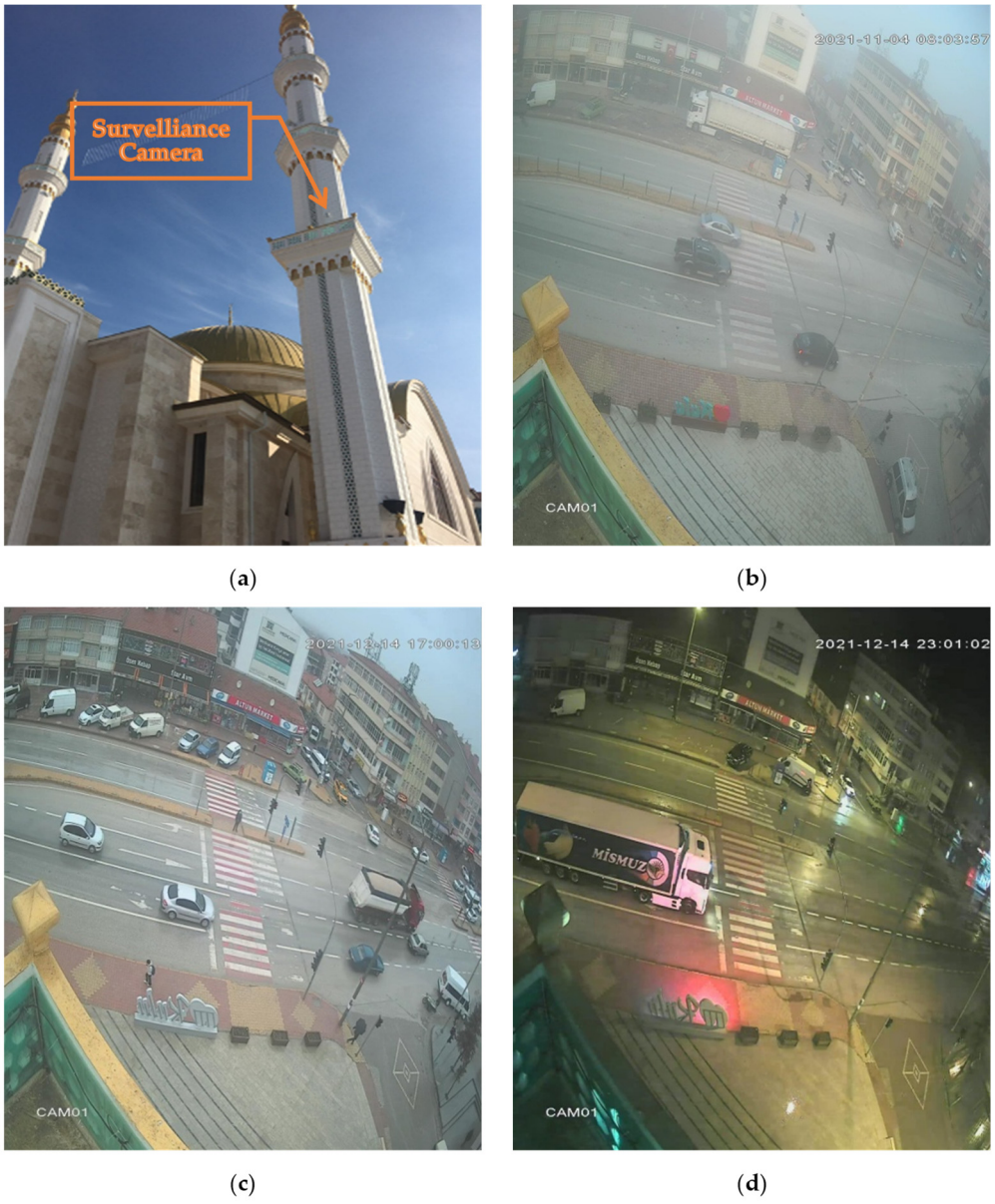
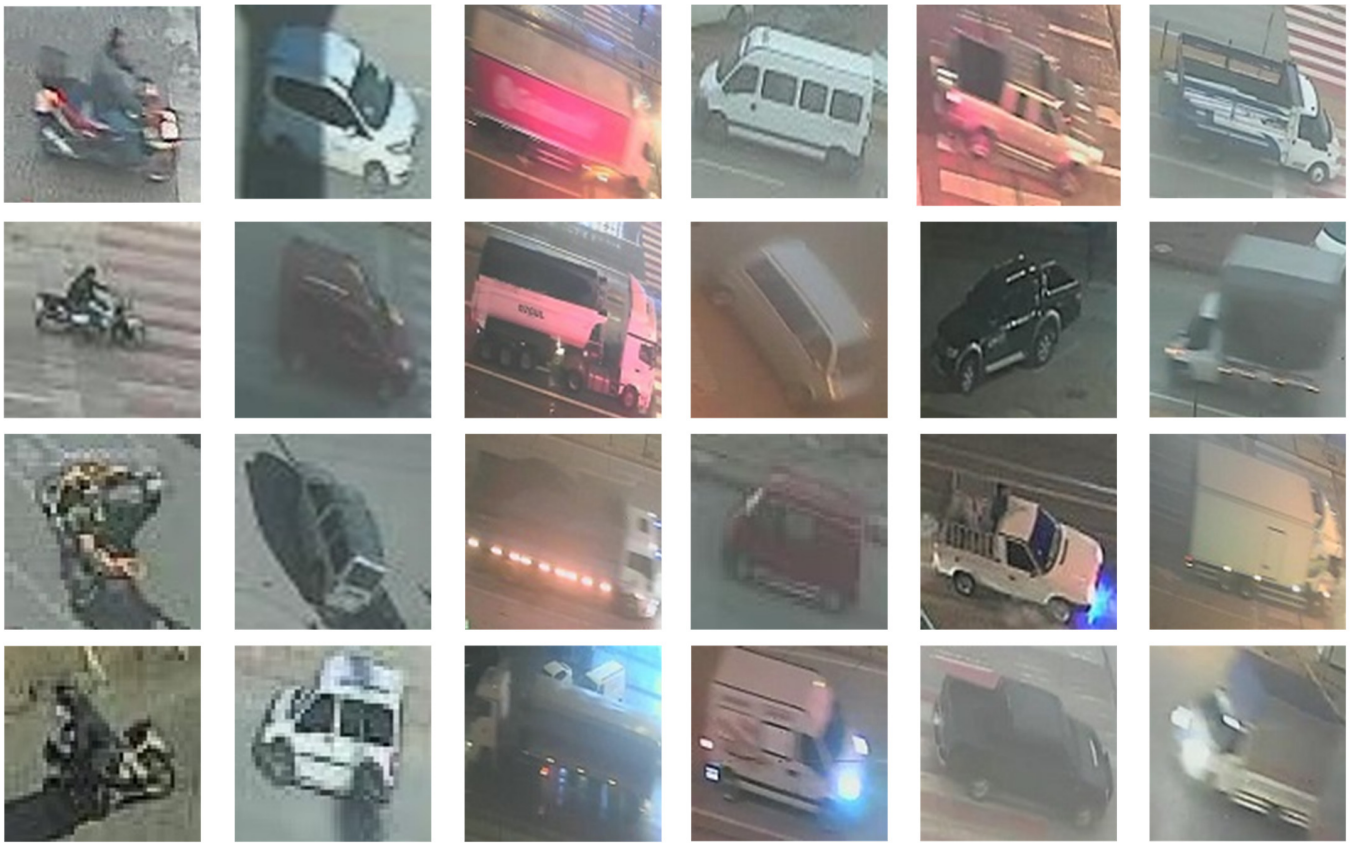


Figure 2. (**a**) The camera mounted on the minaret, (**b**) the view from the camera under daylight haze conditions, (**c**) the view from the camera under daylight rain conditions, (**d**) the view from the camera under nightlight rain conditions.

In the training phase, the RMSProp optimizer was used due to its more stable training performance achieved in initial experiments in comparison to other optimizers such as Adam, Gradient Descent, AdaDelta, and Adagrad. The learning rate of the RMSProp optimizer was set to 0.0001. Moreover, the loss used in the experiments was sparse categorical cross-entropy, since it produces a category index of the most likely matching category. Furthermore, a checkpoint function was implemented with validation accuracy with the monitor and mode set to maximum in order to save the best weights. Additionally, a callback function was used with validation loss as monitor and patience set to 5. The

training was completed in 60 epochs, where the batch size was set to 32, and the dropout layer was set to 0.3. Both the proposed model and the models introduced in Section 5.2 were trained and tested on the same dataset, software, and hardware. The input data were the normalized images, and the target data were the vehicle classes associated with each image.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

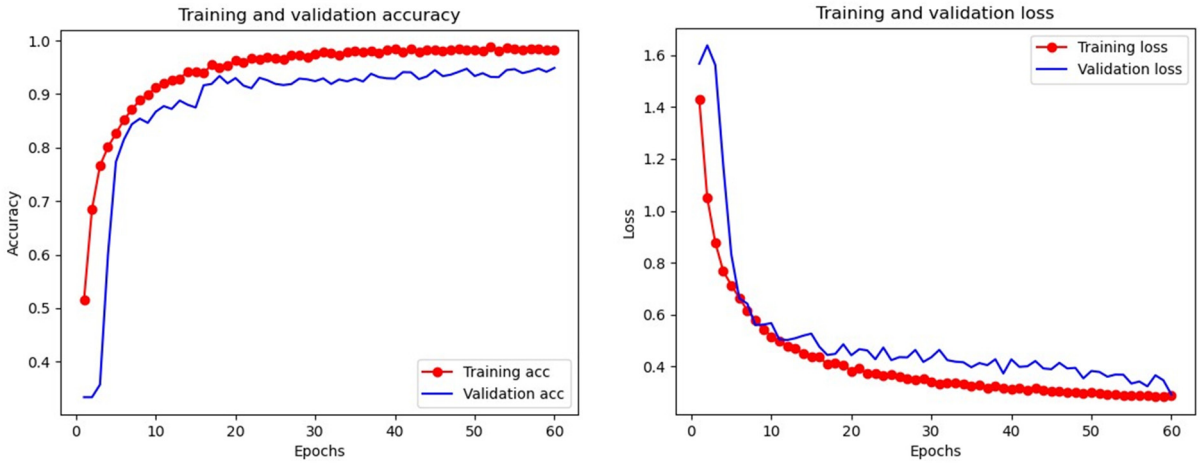
Figure 3. Samples of vehicles: (**a**) bike, (**b**) car, (**c**) juggernaut, (**d**) minibus, (**e**) pickup, (**f**) truck.

5. Results

The results obtained from the experiments are discussed in two steps. In the first step, the effectiveness of the proposed model on the created dataset is evaluated. In the second step, the performance of the proposed model with the well-known CNN-based models is comparatively assessed.

5.1. Performance of the Proposed Model

In order to validate the proposed model on the created dataset, learning curves of the proposed model, namely, the training and validation accuracy and loss curves, were obtained. The overall training and validation accuracy and loss curves of the proposed model are shown in Figures 4a and 4b, respectively. The proposed model achieves a training accuracy of 99.3% with a training loss of 10.4%, and a validation accuracy of 95.5% with a validation loss of 32.0%. From the figures, deviations in the accuracy and loss curves of the proposed model can be easily observed in the first five epochs. However, after epoch 5, the curves converge toward each other. Overall, it is clear that the proposed model is well trained, and overfitting is not observed.



(**a**) (**b**)

Figure 4. Results for the proposed model: (**a**) training and validation accuracy, and (**b**) training and validation loss.

5.2. Comparison with Well-Known CNN Models

The use of CNNs in various applications has been thoroughly studied by researchers in the past few years [38,39]. With the advent of new ideas, powerful hardware and extensive datasets enable the development of various improved models, such as VGG [40], ResNet [41,42], Inception [43], DenseNet [44], MobileNet [45], and Xception [46]. On the other hand, transfer learning is a technique in which a pre-trained model on a similar task is used as a starting point for training a new model on a different task. In this study, the variants of these models, including DenseNet121, ResNet50, Inception v3, Inception-ResNet v2, Extreme Inception (Xception), and VGG19 architectures, were selected to be used as benchmark models due to their higher performances in image classification.

On the other hand, transfer learning is a technique in which a model pre-trained on a similar task is used as a starting point for training a new model on a different task [47]. The well-known CNN models that were used for benchmarking in this study have been developed and highly optimized for specific tasks. Transfer learning gives the option to use the layers of these models for a new task without the need to design a new model and optimize it. In general, transfer learning has several benefits, which include providing better performance, saving training time, and requiring fewer data. Therefore, in this study, transfer learning was used to train benchmarking models on the created dataset.

The comparative performances of the proposed model and the CNN-based models are summarized in Table 2 in terms of overall test accuracy and loss, the number of layers, the parameters used in the architectures, the elapsed training time, the inference time, f1-score, recall, and precision. Moreover, the comparison between the proposed model and the CNN-based models in terms of classification accuracy for each vehicle type is presented in Table 3.

Table 2. Performance comparison between the proposed model and the CNN-based models.

Models	Accuracy (%)	Loss (%)	# Layers	# Parameters (Million)	Training Time (min)	Inference Time (ms)	F1-Score (%)	Recall(%)	Precision (%)
DenseNet121	97.4	22.8	431	~7	27	1.8	97.41	97.11	97.75
ResNet50	97.4	24.7	179	~24	32	1.7	97.4	97.11	97.74
Proposed Model	95.8	30.1	15	~4	9	0.9	95.52	95	96.26
Inception-ResNet v2	95.3	22.2	784	~54	48	2.8	95.03	94.69	95.4
Xception	95.2	25.6	136	~21	35	1.6	94.87	94.3	95.63
VGG19	94.4	47.6	26	~20	49	6.9	93.92	93.67	94.24
Inception v3	93.3	37.3	315	~22	25	1.4	93.19	93.12	93.28

Table 3. Vehicle type based accuracy comparison between the proposed model and the CNN-based models.

Models	Accuracy (%)					
	Bike	**Car**	**Juggernaut**	**Minibus**	**Pickup**	**Truck**
DenseNet121	99.38	99.06	93.33	93.12	92.57	96.68
ResNet50	100	99.06	95.62	92.5	92.35	96.88
Proposed Model	100	98.75	91.22	90.62	93.46	95.62
Inception-ResNet v2	98.75	97.19	93.49	86.88	90.62	95.94
Xception	99.38	96.56	92.54	83.75	90.47	95.4
VGG19	98.12	95.62	89.6	86.25	85.99	93.75
Inception v3	96.88	95.61	96.23	81.12	86.24	91.88

As can be seen in Table 2, with an accuracy of 97.4%, both ResNet50 and DenseNet121 have better overall test accuracy than the other models. Similar results can be achieved when the performances of the models are compared in terms of f1-score, recall, and precision metrics. On the other hand, the proposed model achieved better performance when compared to the Inception v3, Inception-ResNet v2, Xception, and VGG19 models in terms of test accuracy, f1-score, recall, and precision metrics. This is also the case when the results listed in Table 3 are evaluated.

In practice, it is necessary to consider the energy consumption and hardware limitations of IoT devices. Thus, the complexity of the models versus their accuracies needs to be evaluated properly. Although both ResNet50 and DenseNet121 have better classification accuracy than the others, there is a significant drawback stemming from their design space. In Table 2, it is clear that their design space is larger because of the high number of parameters and layers used in their networks. For instance, the ResNet50 model has 179 layers with around 24 Million (M) parameters, whereas the DenseNet121 model has 431 layers with around 7 M parameters.

From Table 2, it is evident that a larger design space adversely affects the training and inference times of the models. Specifically, it took about 32 and 27 min (min) to train the ResNet50 and DenseNet121 models, respectively, while the inference time was about 1.7 and 1.8 milliseconds (ms) for the ResNet50 and DenseNet121 models, respectively. Therefore, there might be some limitations in practice due to the higher computational requirements and resource consumption of these models. At this point, it is obvious that a simple but accurate model needs to be used at the expense of accuracy. When the experimental results are evaluated from this perspective, the efficiency of the proposed model can easily be observed in terms of the number of layers, the trainable parameters, the training time, and the inference time. In comparison to the other models, the architecture of the proposed model has a lower number of layers (15 layers) and the trainable parameters (around 4 M). Additionally, the proposed model is faster because of the fact that the elapsed training time and the inference time of the proposed model were observed to be around 9 min and 1 ms, respectively. Thus, these results suggest that the proposed model could be an efficient alternative to the well-known CNN-based models to be used in IoT-based ITS applications.

The proposed model has a higher loss compared to most of the CNN-based models, as it has fewer layers. Deeper models extract more features from images, leading to lower loss values. However, deeper models also have more parameters, which increases training and inference time. The proposed model is designed to be simple but efficient for IoT devices; hence, it sacrifices some accuracy for speed.

6. Discussion

In general, designing DL models for practical applications requires computational accuracy, model size, and the relationship between the number of trainable parameters and training or inference time. Training and inference time are directly related to the complexity of the model. Models with fewer parameters are expected to achieve faster training and

inference times. In IoT applications, it is necessary to design efficient models with low computational load and minimum memory requirements. Obviously, by minimizing the number of parameters while sustaining high accuracy, models could be used in IoT devices that have limited storage and computing power. In this context, due to its small size architecture and computational efficiency, we believe that the model proposed in this article is suitable for IoT-based ITS applications where resource usage is a significant concern.

The experiments conducted in this study demonstrate that the proposed model is effective in comparison to several well-known CNN-based models in terms of accuracy and complexity on the created dataset [37]. Nevertheless, it is important to note that the proposed model is specifically developed for vehicle classification in low-quality imaging conditions. Even though the proposed model has obvious potential to provide reasonable performance in IoT-based ITS applications, there are still certain concerns that need to be addressed. First of all, the CNN-based model proposed in this study was built to solve a particular classification problem considering low-quality vehicle images captured by imperfectly installed surveillance cameras under poor imaging conditions such as low lighting conditions and various weather conditions. Therefore, the effectiveness of the proposed model was tested on only the created dataset. Since other publicly available datasets contain high-quality images, they were not considered in the assessment of the proposed model's efficiency and performance.

Another concern may be related to the size of the created dataset. In general, CNN-based models have superior performance on large-scale image datasets. The dataset created in this study, on the other hand, is relatively small and contains 4800 images, as mentioned in the previous sections. In fact, small-sized datasets may lead to significant challenges in building an efficient prediction model. In this case, it might be difficult to provide an efficient prediction model that achieves higher accuracy rates [48]. For instance, when the experimental results listed in Table 3 are considered, it can be seen that all the models provide relatively lower classification accuracy for the vehicle types, such as juggernaut, minibus, and pickup. One of the reason for the low accuracy can be described by a high variation in such vehicle types' images, which highly differ in size and the type of load being transported. This may then create confusion with other types of vehicles. Additionally, the presence of advertisement logos on the semi-trailers of some juggernauts could be another reason for the low accuracy. They might divert network attention, which could adversely affect the classification performance. Therefore, in order to achieve more robust results, the dataset size needs to be increased. Nevertheless, the overall experimental results show that the proposed model could achieve acceptable accuracy even with a small-scale image dataset.

Furthermore, similar to [36], a method for vehicle detection from the video frames was not utilized. Instead, the vehicle images were manually cropped from the video frames, which were then saved in a dataset. However, as is already well known, in order to provide efficient vehicle classification, a robust vehicle detection algorithm needs to be employed in a traffic monitoring system. Currently, the authors are focused on the development of a proper vehicle detection algorithm that can be integrated into the classification schemes presented in this study.

7. Conclusions

In this study, a simple CNN-based model for vehicle classification in IoT-based ITS applications was proposed using surveillance images collected by a standard security camera installed far from the traffic scene in low lighting and different weather conditions. The proposed model and other well-known CNN-based models were tested on a newly created dataset. According to the results, with an accuracy of 95.8% (or F1-score of 95% or more), the proposed model achieved better overall classification performance when compared to the Inception v3, Inception-ResNet v2, Xception, and VGG19 models. On the other hand, both the ResNet50 and DenseNet121 models achieved the best performance, with an accuracy of 97.4%. However, the proposed model has advantages in terms of its simple

architecture, which consists of 15 layers, while the ResNet50 and DenseNet121 models use 179 and 431 layers, respectively. Accordingly, the number of trainable parameters is around 4 M, whereas the number of trainable parameters used in the ResNet50 and DenseNet121 models is around 24 M and 7 M, respectively. From the test results, the effects of simpler architecture on the elapsed training times for the created dataset were also observed. While the elapsed training time of the ResNet50 and DenseNet121 models was around 32 min and 2 min, respectively, the training time was found to be around 9 min for the proposed model.

In conclusion, our results verify that the proposed model could be a reasonable alternative for the classification of low-quality vehicle images in poor imaging conditions due to its acceptable accuracy and simple architecture. This makes the proposed method a feasible option to be executed on edge devices in IoT-based ITS applications where their computational capabilities and limited resources are concerned. It is expected that the proposed model could operate effectively in such applications with low computational resources. Particularly, the simple architecture and computational efficiency of the proposed model make it implementable in real-world settings, such as traffic flow monitoring, where a fast response time is crucial to avoid traffic congestion. Therefore, it is believed that this study might offer some new insights into the development of future ITS applications.

In the future, we aim to provide a vehicle detection algorithm that can be employed to extract vehicle images from low-resolution video frames recorded by a low-cost surveillance camera under various imaging conditions. This will enable us to achieve a complete system that can be used in practice. Additionally, this will also enable us to increase the dataset size, which may have an impact on the proposed model's applicability to real-world settings. Nevertheless, the evaluation of the proposed classification model needs to be enhanced by considering additional metrics. In this way, it could be possible to discuss the limitations and strengths of the proposed model for practical applications.

Author Contributions: Data curation, software, B.M.; formal analysis, investigation, B.M., Y.D.; writing—original draft preparation, Y.D.; writing—review and editing, validation, supervision, A.K. and M.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available in a publicly accessible repository. The data presented in this study are openly available on Zenodo at https://doi.org/10.5281/zenodo.8282759, accessed on 26 October 2023.

Acknowledgments: The authors would like to thank Ozgen Sari for his support in data collection.

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

References

1. Bilotta, S.; Collini, E.; Nesi, P.; Pantaleo, G. Short-Term Prediction of City Traffic Flow via Convolutional Deep Learning. *IEEE Access* **2022**, *10*, 113086–113099. [CrossRef]
2. Mandal, V.; Mussah, A.R.; Jin, P.; Adu-Gyamfi, Y. Artificial Intelligence-Enabled Traffic Monitoring System. *Sustainability* **2020**, *12*, 9177. [CrossRef]
3. Hsu, C.-M.; Chen, J.-Y. Around View Monitoring-Based Vacant Parking Space Detection and Analysis. *Appl. Sci.* **2019**, *9*, 3403. [CrossRef]
4. Jang, C.; Kim, C.; Lee, S.; Kim, S.; Lee, S.; Sunwoo, M. Re-Plannable Automated Parking System With a Standalone Around View Monitor for Narrow Parking Lots. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 777–790. [CrossRef]
5. Cui, G.; Zhang, W.; Xiao, Y.; Yao, L.; Fang, Z. Cooperative Perception Technology of Autonomous Driving in the Internet of Vehicles Environment: A Review. *Sensors* **2022**, *22*, 5535. [CrossRef] [PubMed]
6. Claussmann, L.; Revilloud, M.; Gruyer, D.; Glaser, S. A Review of Motion Planning for Highway Autonomous Driving. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1826–1848. [CrossRef]
7. Gholamhosseinian, A.; Seitz, J. Vehicle Classification in Intelligent Transport Systems: An Overview, Methods and Software Perspective. *IEEE Open J. Intell. Transp. Syst.* **2021**, *2*, 173–194. [CrossRef]

8. Won, M. Intelligent Traffic Monitoring Systems for Vehicle Classification: A Survey. *IEEE Access* **2020**, *8*, 73340–73358. [CrossRef]
9. Shokravi, H.; Shokravi, H.; Bakhary, N.; Heidarrezaei, M.; Rahimian Koloor, S.S.; Petrů, M. A Review on Vehicle Classification and Potential Use of Smart Vehicle-Assisted Techniques. *Sensors* **2020**, *20*, 3274. [CrossRef]
10. Maity, S.; Bhattacharyya, A.; Singh, P.K.; Kumar, M.; Sarkar, R. Last Decade in Vehicle Detection and Classification: A Comprehensive Survey. *Arch. Comput. Methods Eng.* **2022**, *29*, 5259–5296. [CrossRef]
11. Hussain, K.F.; Afifi, M.; Moussa, G. A Comprehensive Study of the Effect of Spatial Resolution and Color of Digital Images on Vehicle Classification. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 1181–1190. [CrossRef]
12. Pavel, M.I.; Tan, S.Y.; Abdullah, A. Vision-Based Autonomous Vehicle Systems Based on Deep Learning: A Systematic Literature Review. *Appl. Sci.* **2022**, *12*, 6831. [CrossRef]
13. Dong, Z.; Wu, Y.; Pei, M.; Jia, Y. Vehicle Type Classification Using a Semisupervised Convolutional Neural Network. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2247–2256. [CrossRef]
14. Yu, S.; Wu, Y.; Li, W.; Song, Z.; Zeng, W. A Model for Fine-Grained Vehicle Classification Based on Deep Learning. *Neurocomputing* **2017**, *257*, 97–103. [CrossRef]
15. Zhuo, L.; Jiang, L.; Zhu, Z.; Li, J.; Zhang, J.; Long, H. Vehicle Classification for Large-Scale Traffic Surveillance Videos Using Convolutional Neural Networks. *Mach. Vis. Appl.* **2017**, *28*, 793–802. [CrossRef]
16. Chang, J.; Wang, L.; Meng, G.; Xiang, S.; Pan, C. Vision-Based Occlusion Handling and Vehicle Classification for Traffic Surveillance Systems. *IEEE Intell. Transp. Syst. Mag.* **2018**, *10*, 80–92. [CrossRef]
17. Maungmai, W.; Nuthong, C. Vehicle Classification with Deep Learning. In Proceedings of the 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), Singapore, 23–25 February 2019; pp. 294–298.
18. Wang, X.; Zhang, W.; Wu, X.; Xiao, L.; Qian, Y.; Fang, Z. Real-Time Vehicle Type Classification with Deep Convolutional Neural Networks. *J. Real-Time Image Process.* **2019**, *16*, 5–14. [CrossRef]
19. Jahan, N.; Islam, S.; Foysal, M.F.A. Real-Time Vehicle Classification Using CNN. In Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 1–3 July 2020; pp. 1–6.
20. Sharma, P.; Singh, A.; Singh, K.K.; Dhull, A. Vehicle Identification Using Modified Region Based Convolution Network for Intelligent Transportation System. *Multimed. Tools Appl.* **2022**, *81*, 34893–34917. [CrossRef]
21. Ma, Z.; Chang, D.; Xie, J.; Ding, Y.; Wen, S.; Li, X.; Si, Z.; Guo, J. Fine-Grained Vehicle Classification with Channel Max Pooling Modified CNNs. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3224–3233. [CrossRef]
22. Chauhan, M.S.; Singh, A.; Khemka, M.; Prateek, A.; Sen, R. Embedded CNN Based Vehicle Classification and Counting in Non-Laned Road Traffic. In Proceedings of the Tenth International Conference on Information and Communication Technologies and Development, Ahmedabad, India, 4–7 January 2019; pp. 1–11.
23. Hasan, M.M.; Wang, Z.; Hussain, M.A.I.; Fatima, K. Bangladeshi Native Vehicle Classification Based on Transfer Learning with Deep Convolutional Neural Network. *Sensors* **2021**, *21*, 7545. [CrossRef]
24. Mittal, U.; Potnuru, R.; Chawla, P. Vehicle Detection and Classification Using Improved Faster Region Based Convolution Neural Network. In Proceedings of the 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 4–5 June 2020; pp. 511–514. [CrossRef]
25. Hassan, R.; Qamar, F.; Hasan, M.K.; Aman, A.H.M.; Ahmed, A.S. Internet of Things and Its Applications: A Comprehensive Survey. *Symmetry* **2020**, *12*, 1674. [CrossRef]
26. Derawi, M.; Dalveren, Y.; Cheikh, F.A. Internet-of-Things-Based Smart Transportation Systems for Safer Roads. In Proceedings of the 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 2–16 June 2020; pp. 1–4.
27. Biswas, A.; Wang, H.-C. Autonomous Vehicles Enabled by the Integration of IoT, Edge Intelligence, 5G, and Blockchain. *Sensors* **2023**, *23*, 1963. [CrossRef]
28. Chmaj, G.; Lazeroff, M. IoT Machine Learning Based Parking Management System with Anticipated Prediction of Available Parking Spots. In Proceedings of the ITNG 2022 19th International Conference on Information Technology-New Generations, Las Vegas, NV, USA, 10–13 April 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 341–350.
29. Babu, K.R.M. IOT for ITS: An IOT Based Dynamic Traffic Signal Control. In Proceedings of the 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 11–12 July 2018; pp. 532–537.
30. Bautista, C.M.; Dy, C.A.; Mañalac, M.I.; Orbe, R.A.; Cordel, M. Convolutional Neural Network for Vehicle Detection in Low Resolution Traffic Videos. In Proceedings of the 2016 IEEE Region 10 Symposium (TENSYMP), Bali Island, Indonesia, 9–11 May 2016; pp. 277–281.
31. Chen, W.; Sun, Q.; Wang, J.; Dong, J.-J.; Xu, C. A Novel Model Based on AdaBoost and Deep CNN for Vehicle Classification. *IEEE Access* **2018**, *6*, 60445–60455. [CrossRef]
32. Roecker, M.N.; Costa, Y.M.; Almeida, J.L.; Matsushita, G.H. Automatic Vehicle Type Classification with Convolutional Neural Networks. In Proceedings of the 2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP), Maribor, Slovenia, 20–22 June 2018; pp. 1–5.
33. Tsai, C.-C.; Tseng, C.-K.; Tang, H.-C.; Guo, J.-I. Vehicle Detection and Classification Based on Deep Neural Network for Intelligent Transportation Applications. In Proceedings of the 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Honolulu, HI, USA, 12–15 November 2018; pp. 1605–1608.
34. Wang, X.; Chen, X.; Wang, Y. Small Vehicle Classification in the Wild Using Generative Adversarial Network. *Neural Comput. Appl.* **2021**, *33*, 5369–5379. [CrossRef]

35. Butt, M.A.; Khattak, A.M.; Shafique, S.; Hayat, B.; Abid, S.; Kim, K.-I.; Ayub, M.W.; Sajid, A.; Adnan, A. Convolutional Neural Network Based Vehicle Classification in Adverse Illuminous Conditions for Intelligent Transportation Systems. *Complexity* **2021**, *2021*, 6644861. [CrossRef]
36. Tas, S.; Sari, O.; Dalveren, Y.; Pazar, S.; Kara, A.; Derawi, M. Deep Learning-Based Vehicle Classification for Low Quality Images. *Sensors* **2022**, *22*, 4740. [CrossRef] [PubMed]
37. Maiga, B.; Dalveren, Y.; Kara, A.; Derawi, M. A Dataset Containing Tiny Vehicle Images Collected in Low Quality Imaging Conditions. Available online: https://zenodo.org/records/8282760 (accessed on 23 November 2023).
38. Ghimire, D.; Kil, D.; Kim, S. A Survey on Efficient Convolutional Neural Networks and Hardware Acceleration. *Electronics* **2022**, *11*, 945. [CrossRef]
39. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 6999–7019. [CrossRef]
40. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:14091556.
41. Shafiq, M.; Gu, Z. Deep Residual Learning for Image Recognition: A Survey. *Appl. Sci.* **2022**, *12*, 8972. [CrossRef]
42. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
43. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, Boston, MA, USA, 7–12 July 2015; IEEE: New York, NY, USA, 2015; pp. 1–9.
44. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, Honolulu, HI, USA, 21–26 July 2017; IEEE: New York, NY, USA, 2017; pp. 4700–4708.
45. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv170404861.
46. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, Honolulu, HI, USA, 21–26 July 2017; IEEE: New York, NY, USA, 2017; pp. 1251–1258.
47. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A Comprehensive Survey on Transfer Learning. *Proc. IEEE* **2021**, *109*, 43–76. [CrossRef]
48. Kim, P. *MATLAB Deep Learning*; Machine Learning, Neural Networks and Artificial Intelligence; Apress: Berkeley, CA, USA, 2017; ISBN 978-1-4842-2845-6. [CrossRef]

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.