applied sciences

MDPI

Article

Performance Analysis of Deep Convolutional Network Architectures for Classification of Over-Volume Vehicles

S. Sofana Reka [1], Venkata Dhanvanthar Murthy Voona [2], Puvvada Venkata Sai Nithish [2], Dornadula Sai Paavan Kumar [2], Prakash Venugopal [2,*] and Visvanathan Ravi [2]

[1] Centre for Smart Grid Technologies, School of Electronics Engineering, Vellore Institute of Technology, Chennai 600127, India
[2] School of Electronics Engineering, Vellore Institute of Technology, Chennai 600127, India
* Correspondence: prakash.v@vit.ac.in

Abstract: The number of vehicle accidents has increased in recent years due to overloaded goods carriers. Off-road driving, mountain roads, and sharp edges on a road are the main causes of an imbalance in overloaded trucks. In rural areas, where smaller roads cannot accommodate high volume vehicles, such vehicles cause many problems for cars, bikes, bicycles, and other small vehicles, as well as an increase in traffic congestion in those areas. This has become a major problem in the daily lives of drivers in rural areas as well as major urban areas. Solutions are needed to detect over-volume goods carriers and alert drivers to slow down or control the volume in their trucks. This work mainly focuses on a solution that uses deep CNN models. In this work, different deep convolutional architectures are evaluated for their ability to classify goods based on their volume. The model implemented is based on a dataset-specific transfer learning process with CNN layers generated in ImageNet in which only dense layers are learned. The primary objective of this work is to identify a classification method that exhibits proven results with respect to the accuracy parameters. In this work, different deep architectures were tested, and among the efficient networks, Net-B3 was found to perform with 95% accuracy on average. The different architectures were evaluated based on their accuracy, confusion matrix, ROC curve, and AUC score with a real-time dataset.

Keywords: over-volume goods carriers; convolutional neural networks; deep learning; transfer learning; fully connected layers; performance metrics

check for updates

Citation: Reka, S.S.; Murthy Voona, V.D.; Sai Nithish, P.V.; Paavan Kumar, D.S.; Venugopal, P.; Ravi, V. Performance Analysis of Deep Convolutional Network Architectures for Classification of Over-Volume Vehicles. *Appl. Sci.* **2023**, *13*, 2549. https://doi.org/10.3390/app13042549

Academic Editors: Daniele Spoladore, Alberto Trombetta and Andrea Zangiacomi

Received: 18 December 2022
Revised: 8 February 2023
Accepted: 10 February 2023
Published: 16 February 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

It has become increasingly common for vehicles to be overloaded, leading to accidents. As a result of overloading, a driver's ability to brake and steer is reduced, which can lead to accidents. With respect to the unnecessary stress on the engine, the chances of tire failure increase at a greater rate, which reduces vehicle stability [1,2]. Road accidents caused by vehicles transporting goods are on the rise due to overloading. The other major concern is the possibility of casualties that may result from these truckloads exceeding their capacity. This may also cause damage to roads to a greater extent. As per the work mentioned in [1,2], difficulties in meeting capacity requirements are caused by errors in design and implementation. Overloading also has an effect on the uneven maintenance and repair of roads, as is mentioned in [3–5].

In this regard, truck users who load beyond the maximum capacity can cause major concern with a realistic probability of accidents on highways [6]. Analysis has shown that these heavy overloaded trucks are predominantly found near factory areas, leading to these roads breaking down more quickly. In addition, the need for road repairs causes concern every year given their serious effects on road users. From a wider perspective, these situations disrupt the economy of the nation. Deeper analysis shows that overloaded vehicles damage the road and occupy most of the space on the road, especially in rural

areas. A major concern for rural areas is that most of the commuting roads in India are single- and double-lane only. In addition to these adverse drawbacks, it is difficult for other drivers to overtake these vehicles. Another problem with vehicles overcrowded with goods is the possibility of disturbing power lines and properties near the road. As we are moving towards the era of smart cities in developing countries, solutions for this problem are to be identified at the preliminary stages using machine learning. Solutions to overcome this problem can be achieved with a deep convolutional neural network for classifying overloaded and non-overloaded vehicles that can be used to detect overloaded vehicles in the absence of human interference. This shows the way to solve this issue with respect to the global standardization of the smart city. In the field of computer vision, deep learning is the most effective algorithm for solving problems, in part due to its capacity for object detection and classification, which exhibit the best solutions for this problem. Thereby, this model allows for a complete check on over-volume vehicles in areas where human supervision may not be possible, such as in remote mountain regions, sub-rural areas, and forest areas. In this way, this model provides a solution for the classification of over-loaded vehicles.

This work proposes the best performing model for identifying over-volume vehicles applied in the setting of roads and traffic. In this work, we use the ImageNet weights of complex DCNNs as the initial features, perform fine-tuning by unfreezing one-fourth of the portion of the neural network, and train the collected dataset. These approaches helped to generate the features of DCNNs related to our collected dataset and significantly increased the performance of DCNNs by 5–10%.

The main contributions of this work are as follows:

- A new real-time dataset consisting of 3054 images categorized into three different classes of over-volume goods carriers, goods carriers that are not over-volume, and vehicles not carrying goods.
- The implementation of various state-of-the-art (SOTA) deep convolutional neural networks for classifying vehicles with the help of transfer learning and fine-tuning.
- A thorough performance analysis of DCNNs on the collected dataset.

2. Related Work

The majority of the existing research focuses on overloaded vehicle detection with the help of sensors and other external devices. In [7], the authors discuss the ground vibrations generated by vehicles with different weights. This work uses an external device called a geophone that helps to measure tiny vibrations in the ground. With the vehicles passing by the road, the geophone can measure the vibrations and transmit the vibration data through the wireless transmission module. In this analysis, a support vector machine algorithm is used on the collected data to detect the overloaded vehicles. The work conducted in [7,8] prioritizes the practical conditions of the vehicle load control framework through the load cell strain size implemented in the vehicle. The single-chip microcontroller gathers data that are transmitted and obtained through sensors and calculates the overall vehicle load.

The work proposed in [9] uses load cells to measure the weights of the goods and introduces an engine locking system in the presence of an overload. The weights measured by load cells are compared to the threshold weights by Arduino HX711, and in the presence of overload the DC motor system's ignition is stopped. Similarly, in this work, the authors introduce a vehicle load control framework with the help of a single-chip microcontroller and weight sensors. In the work proposed in [10], the authors introduce a vehicle load control framework with the help of a single-chip microcontroller and weight sensors. Similarly, in [10], weight sensors are used to determine the entire weight of the load being carried by the vehicle. When the load is too heavy, the microcontroller sends a signal to the vehicle system to prohibit it from starting. Although there are systems that focus on overload vehicle identification, to the best of our knowledge, this is the first study that focuses on classifying over-volume vehicles as a vision-based task. The conventional methods for handling vision-based tasks involve customized feature extractors followed by

machine learning classifiers. For example, categorizing the vehicles in images taken from surveillance cameras as mentioned in [11] uses a histogram of oriented gradients (HOG) descriptor for feature extraction and a support vector machine classifier for classification. Here, they divide the vehicles into three main categories: motorcycles, compact cars, and large cars.

In the past few years, deep convolutional neural networks have become impeccable at handling vision-based tasks such as image classification, object detection, image segmentation, etc., with the help of its high-performing feature extractors. The study proposed in [12] uses deep convolutional neural networks (DCNNs) to classify vehicles in a natural setting without cropping the region of interest of the vehicles. In [13], DCNNs are used for the classification of vehicles into four different categories including school, bus, police, and ambulance. The proposed method achieved an average precision and recall of 89% and 83%, respectively. The authors of [14] used ResNet50 architecture to perform transfer learning to classify thirteen different Bangladeshi traffic vehicle types. The research proposed a new dataset consisting of 10,440 images categorized into thirteen types of vehicles. Similar studies [15] have proposed various state-of-the-art DCNNs and performed fine-tuning on ImageNet [16] weights after transfer learning. ResNet-152 architecture provided the best performance with an accuracy of 98.77%.

The research was performed on a public dataset called 'Poribohon-BD' [17], which consists of a total of 8138 Bangladeshi common vehicle images categorized into 15 types. In [18], the authors collected a new dataset of 10,000 images extracted from road surveillance and driving videos. The dataset is categorized into six classes including rickshaw, motorbike, truck, van, bus, and car. The ResNet architecture gave the best performance with an accuracy of 95.14%. It can be seen that many solutions have been introduced as discussed to classify and identify overloaded vehicles in [7–17]. The authors of [18] showcase great performance when classifying real-time data that consist of different types of vehicles, and this provides us with the motivation to classify over-volume vehicles using deep CNN models. In this respect, few studies have been carried out exhibiting the solution to this problem in the past. The major aim of this work is to develop a realistic new dataset with different categories and to implement the datasets in various state-of-the-art CNN models for classification based on a transfer learning approach and fine-tuning.

3. Materials and Methods

The introduction of deep convolutional neural networks (DCNNs) has created a huge impact, especially in the field of computer vision, due to its exceptional performance at image classification, object detection, and other computer vision tasks. Considering the complex structure of DCNNs, implementing them without over-fitting and convergence issues from the beginning holds a challenge with respect to small-scale training data. These challenges can be overcome by implementing the following methodologies. This section clearly exhibits the real-time dataset obtained and is a requirement of transfer learning approaches.

3.1. Pre-Trained CNN Models and Transfer Learning

Deep CNN architectures are trained on datasets containing a large number of annotated images and produce usable feature extractors that are also called model weights. These weights are mainly used while training on small-scale datasets to avoid over-fitting and reduce the training time. In [19,20], the authors demonstrated that generic features extracted from pre-trained CNN models are very productive for image classification, object detection, and localization from natural images. To improve the performance of a model trained on a specific domain, transferring the information from a model trained on a similar domain of interest is known as transfer learning. With the use of this model, small datasets have shown better results with the use of ImageNet as mentioned in [19,20]. Experimental results from [21,22] also show that deep CNN-based transfer learning attains better classification results with small datasets compared with knowledge learned from other related

tasks with immense datasets. In [23], the authors replaced fully connected layers with logistic layers in a pre-trained classifier CNN model. In the process of model training, the work shows that all the layers in the model were frozen except for the logistic regression layers. This procedure provided improved performance for categorizing multi-view mammograms at a larger aspect, which was proposed in this model for classification aspects.

3.2. Fine-Tuning

To exhibit the method's better performance and overcome its challenges, an advanced fine tuning technique is used, and this process is exhibited after the process of transfer learning where a fraction of CNN layers are unfrozen and the weights are made accessible to train along with features from the fully connected layers.

These weights are trained subject to the required field of interest to make the pre-trained model more relevant to the real-time dataset collected. In [24], the authors unfroze all the required CNN layers of the pre-trained architecture and trained every weight using their specified dataset to distinguish interstitial lung diseases. This understanding shows the importance of the training and fine-tuning of a particular dataset. Experimental results from [25,26] illustrate that the consistent usage of fine tuning has empowered the performance of pre-trained CNN models and produces better results with the same datasets. The successful usage of fine-tuning can be observed in [27,28]. The details of the number of layers unfrozen and trained can be observed in Table 1. Unfreezing the complete neural network and training it on small-scale dataset will lead to overfitting the model. Considering the depth and complexity of the state-of-the-art models, we chose to unfreeze 1/4th of the neural network and to train these layers on the collected dataset, whereas the frozen layers have the weights set on the Imagenet dataset.

Table 1. Fine-tuning of deep CNN models.

S. No.	Model	Tot. No. of Layers	Unfrozen Layers
1	MobileNetV2	158	120
2	ResNet50	178	134
3	VGG19	25	18
4	EfficientNetB0	240	180
5	EfficientNetB3	387	290
6	EfficientnetB4	477	360

Total No. of layers—Total number of layers consisting of input, zero padding, convolutional, and fully connected layers. Unfrozen Layers—Each model is trained from this nth layer to the fully connected layer to predict the output.

The proposed methodology begins with data collection and data augmentation of images related to goods carriers followed by data splitting into train, test, and validation sets. In the subsequent stage, transfer learning was implemented on the state-of-the-art DCNNs where the convolutional layers were frozen with ImageNet weights and only the classifier layers (i.e., fully connected layers) were trained on the collected dataset. Furthermore, fine-tuning was carried out by unfreezing the one-fourth portion of the neural network for the model training. Later, the best performing model was chosen based on different performance metrics. Figure 1 provides a visual overview of the proposed system.

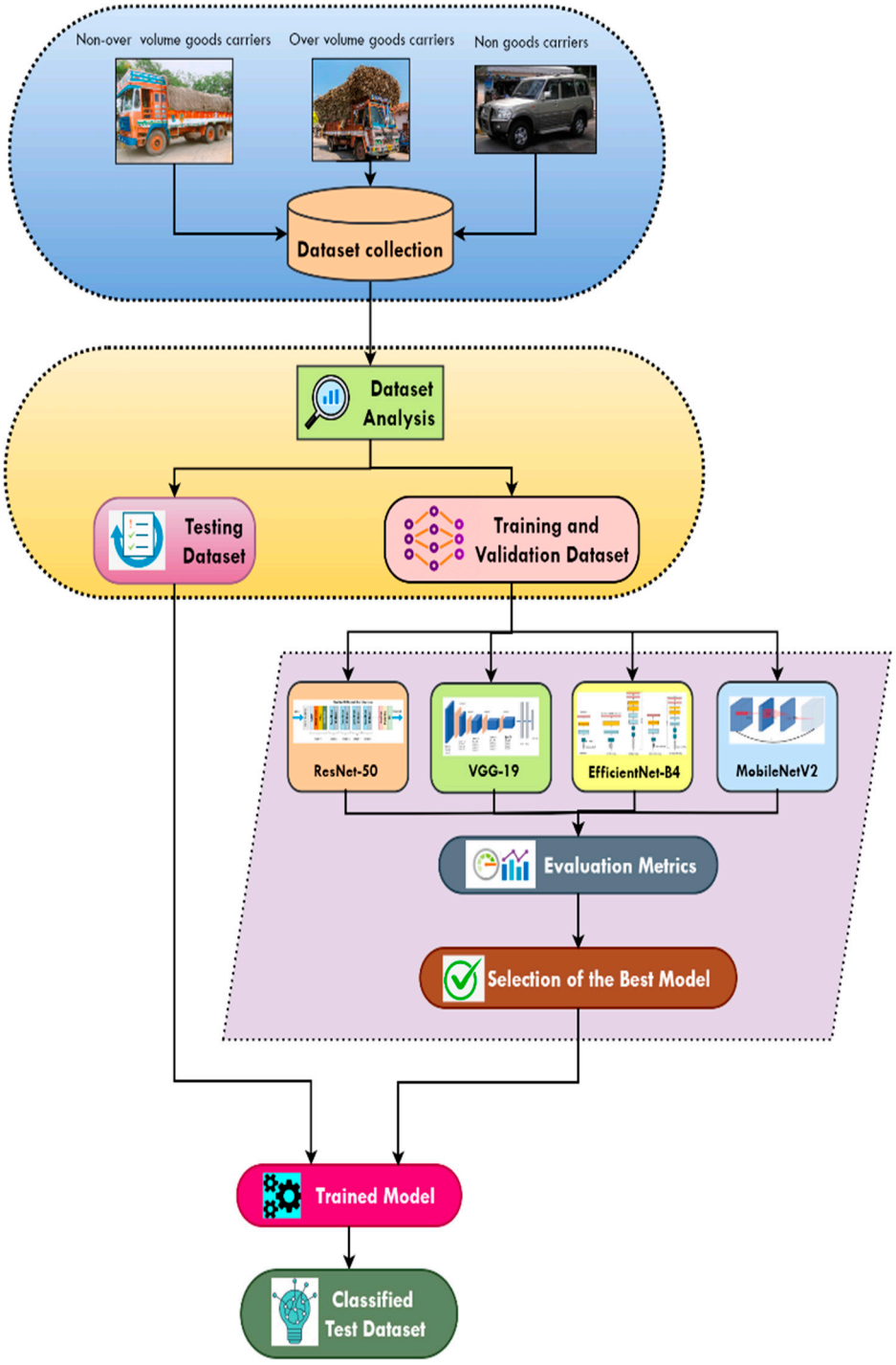


Figure 1. Proposed architecture.

3.3. Materials

The dataset was manually collected through real-time capture, and 20% of the dataset was generated by data augmentation with various functions such as horizontal flipping, warping, and introduction of noise. The real-time dataset was captured based on these features for data augmentation at a larger scale. This helps us to introduce variability to the dataset. Dataset analysis involves data from each class being equally distributed to avoid biasing the model. The dataset collected is in jpg format. In this work, to accomplish the solution, a total of 3054 images were collected. In this collected dataset, three classes were labeled: over-volume goods carriers, non-over-volume goods carriers, and non-goods carrier vehicles. There are approximately 1000 images for each class across both the training

and testing sets. A validation split ratio of 8:1:1 was used, and a detailed description of the data distribution is available in Table 2. The images will be transformed into the required dimensions for each of the state-of-the-art models that are implemented in this work. Table 3 provides a preview of the abovementioned classes that were collected for this work.

Table 2. Detailed description of the dataset consisting of three different classes.

Categories	Training	Testing	Validation	Total
Non-over-volume goods carriers	839	106	104	1049
Over-volume goods carriers	804	101	100	1005
Non-goods carriers	800	100	100	1000
	2443	307	304	3054

Table 3. Preview of images belonging to their respective classes.

		
Non-over-volume goods carrier	Over-volume goods carrier	Non-good carrier

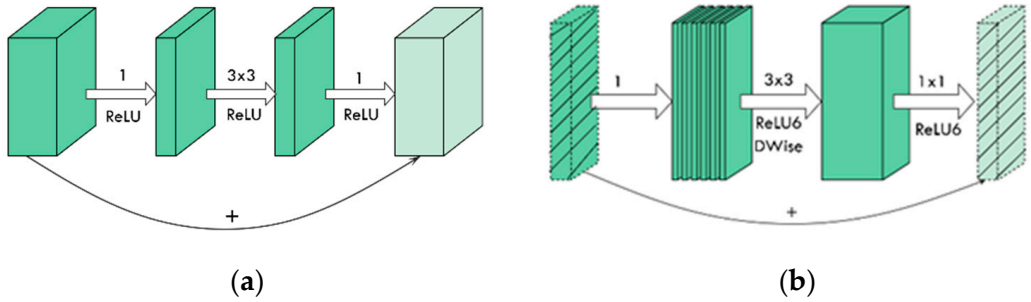
3.4. Methods

This work examines and analyzes the state-of-the-art models of the proposed collected dataset with various changes in the hyper-parameters. VGGNet19, ResNet50, MobileNetV2, and EfficientNet [B0][B3][B4] are the models used in this work for its successful implementation.

3.4.1. MobileNet

MobileNetV2, developed by Google, is a deep c convolutional neural network (DCNN) with enhanced performance and efficiency. The key feature of MobileNetV2 is the introduction of a new structure called inverted residuals [29]. Deep neural networks face the problem of destroying information in non-linear layers in convolutional blocks. The inverted residual bottleneck layers are used to address this problem by preserving all the useful information. The name "MobileNet" suggests its usability in mobile applications. The inverted residual bottleneck layers are particularly memory-efficient, which makes this the best fit for mobile applications.

The initial convolutional layer of MobileNetV2 consists of 32 filters, followed by 19 bottleneck layers, resulting in an increase in the number of filters from 32 to 1280. In order to prevent overfitting, the model uses dropout and batch normalization during training and has a kernel size of 3×3. The dimensions of the input images were resized to 224×224 pixels before being passed to the network with the help of the image data generator class available in the Keras library. The pictorial overview of convolutional blocks in MobileNetV2 is presented in Figure 2.



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

Figure 2. In (**a**), the residual is used to connect the layers with a high number of channels, whereas in (**b**) the bottlenecks are connected by inverted residuals [29].

3.4.2. VggNet

Input images with the dimensions 224 × 224 are used to train the VGG architecture, and the generated weights are used to test the model. As is shown in Figure 3, the kernels increase in each stage as we go deeper into the network from the top [30]. Rectified linear units were used as an activation function for feature learning in both convolutional layers and fully connected layers except the final dense layer. This layer used Softmax as an activation function to classify the three output classes. In back propagation, Adam was used as an optimization algorithm, and categorical cross-entropy was used as a loss function.

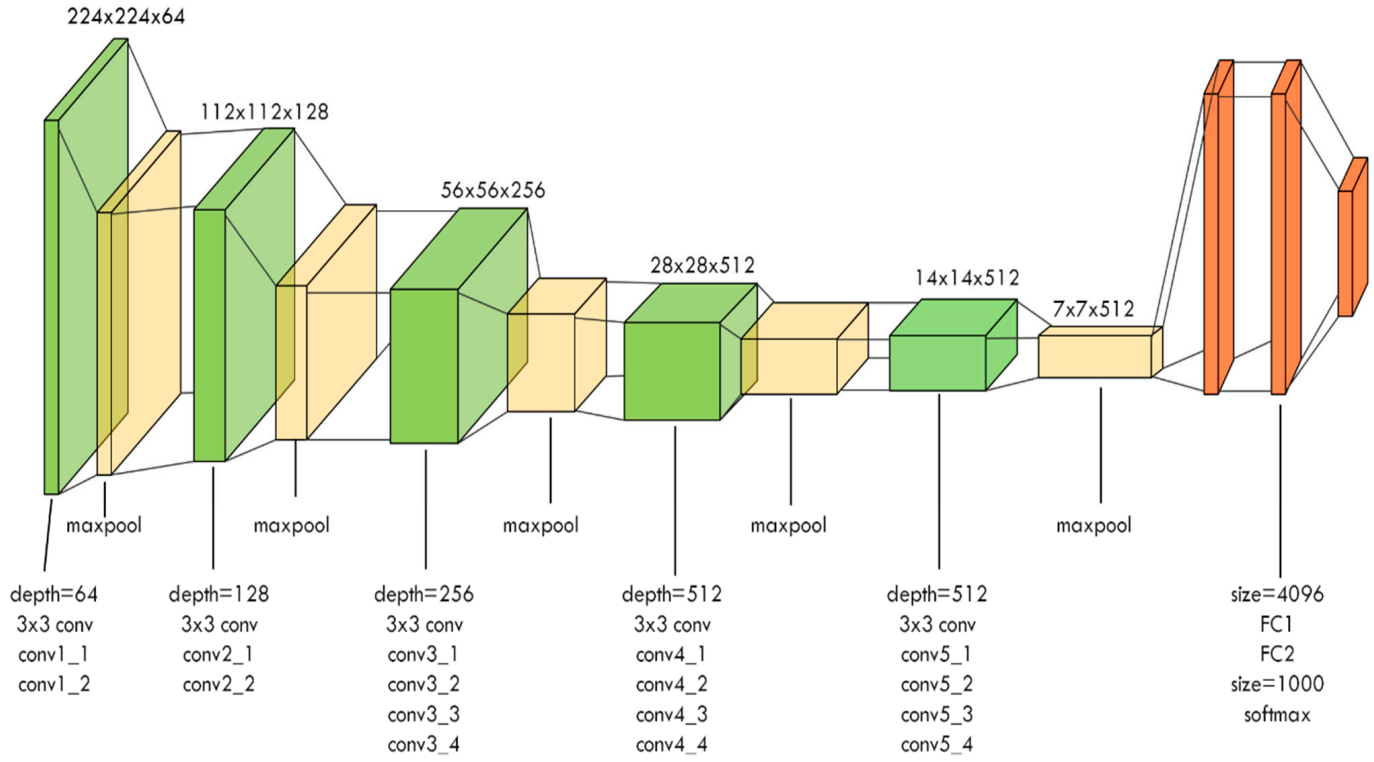


Figure 3. VGG-19 network architecture: conv stands for convolution, FC stands for fully connected [30].

3.4.3. EfficientNet

EfficientNet is a widely used convolutional neural network architecture that was developed by Google Research, Brain Team. The key feature of EfficientNet architecture is the introduction of a new scaling method called compound scaling. This method carefully balances the network width, depth, and resolution to achieve better accuracy and efficiency. Different scaling methods are compared and explained in Figure 4. The researchers started with a baseline model called EfficientNet [B0]. The baseline model uses inverted bottleneck

residual blocks of MobileNetV2 [31], in addition to squeeze-and-excitation blocks. The baseline network is then scaled up to obtain a family of models called EfficientNets. We used EfficientNet [B0], [B3], and [B4] to develop an image classifier in this work. All three architectures use Softmax as their activation function and Adam as their optimization algorithm. The convolution layer weights, which are generated by the ImageNet data, have been frozen, and dense layers' weights were trained by the specific mentioned dataset.

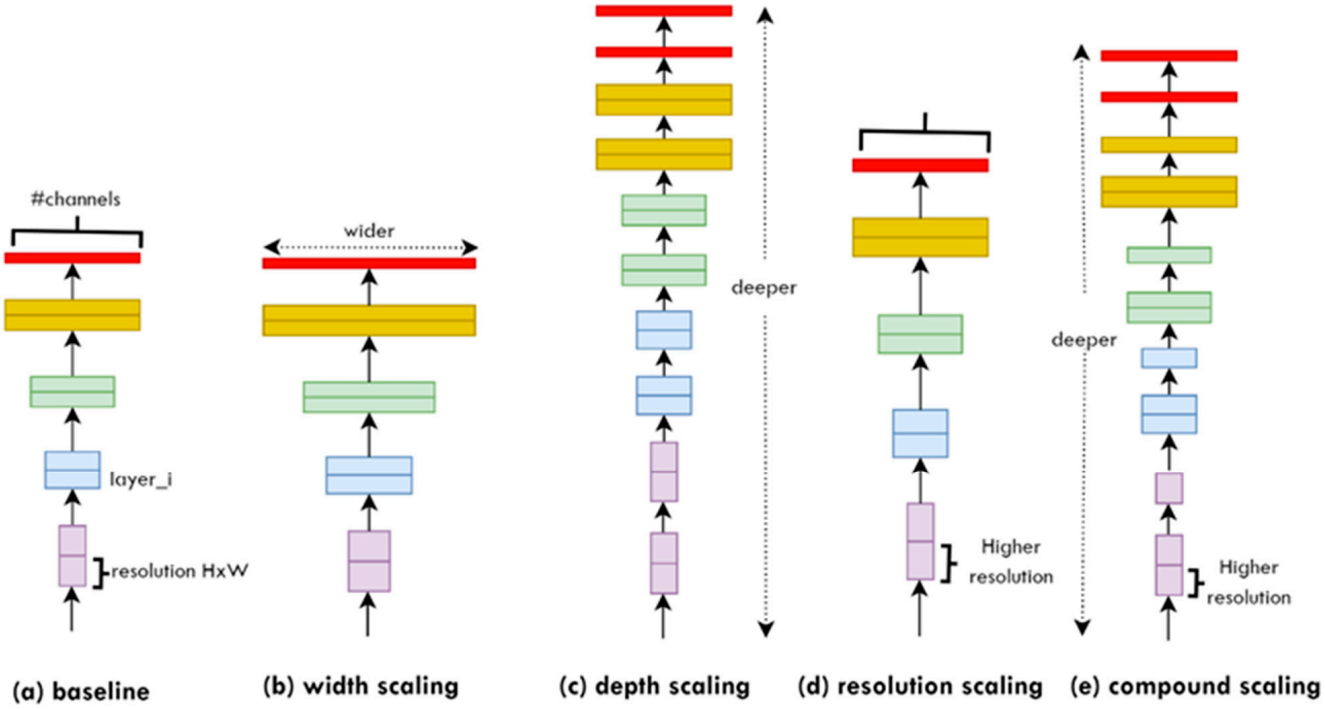


Figure 4. Comparison of different scaling methodologies (**a**–**e**). In contrast to conventional scaling methods (**b**–**d**), which arbitrarily scale a single dimension of the network, the compound scaling method uniformly scales up all dimensions in a principled way [32].

3.4.4. ResNet

A residual network (ResNet) was the winner of the ImageNet challenge in 2015. The reason behind its fine performance was the introduction of 'skip connections' in its architecture [33]. In the back propagation phase, the deep neural networks encounter a problem of vanishing gradients as the gradients near the inputs are obtained by multiplying the gradient of later layers, which is a major concern in depth-scaled deep neural networks. This affects the convergence of the model. To address this problem, ResNet architecture is introduced, which consists of convolutional and identity blocks with skip connections built into it. In the backpropagation phase, these connections help to skip certain layers, which helps reduce the effect of the vanishing gradient problem. Figure 5 gives a visual overview of the skip connections. The final dense layer of this architecture was modified to fit three classes with the help of the Softmax activation function and trained from scratch with our dataset.

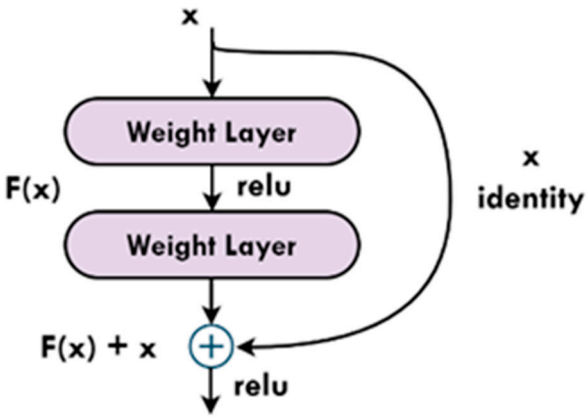
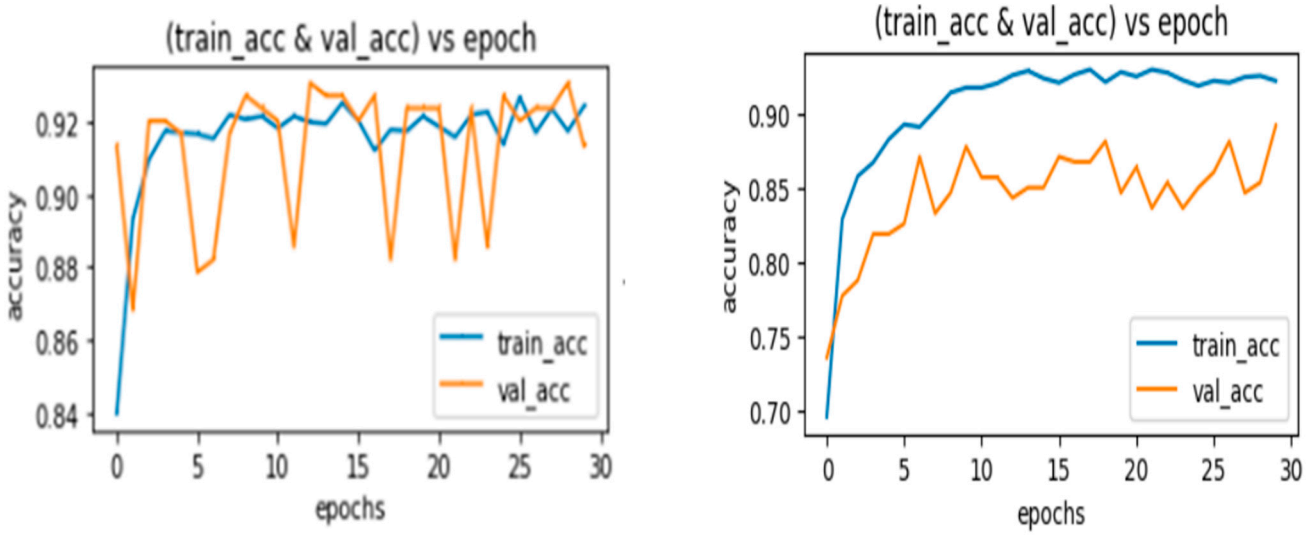


Figure 5. Skip connection (or residual block) as given by He et al., 2017 [29]. Other models train on the output 'Y', but ResNet trains on F(X). ResNet tries to make F(X) = 0 so that Y = X.

3.5. Hyper-Parameter Adjustment

Batch size is an important hyper-parameter in neural network training because it determines how many samples are learned in an iteration before the system updates its weights. A very small batch size may lead to the overfitting of the data, and a high batch size requires advanced GPU support. Thus, as per the need of the model for training, we proceeded with a medium batch size of 32 and trained the models discussed above. The learning rate is another crucial hyperparameter in machine learning training. A high learning rate will result in fluctuating loss function weights after each iteration. In contrast, if the learning rate is set too low, the convergence rate is too slow. We trained the model with an initial learning rate of 0.001 and observed fluctuations in the loss function. In this work, in order to achieve the solution variable learning rate scheme, we implemented an initial learning rate of 0.0001 and after iterations with constant validation accuracy, the learning rate decayed by a factor of two with the patience of two epochs. Figure 6 shows a comparison of the training and validation accuracy of the MobileNetV2 architecture for the learning rates 0.001 and 0.0001. The loss is completely abnormal when the learning rate is 0.001.



(**a**) Learning rate: 0.003. (**b**) Learning rate: 0.0004.

Figure 6. Comparison of different learning rates for the MobileNetV2 model.

4. Results

Several performance metrics were used to scrutinize the performance of the DCNN models, and these metrics are discussed in this section. The Keras library was used to create the deep learning models, and tensor flow served as the backend. The DCNNs were trained on the collected real-time data to classify them into three categories, namely non- over-volume goods carriers, over-volume goods carriers, and non-goods-carriers. On the basis of major criteria such as ROC curves, AUC scores, confusion matrixes, and test accuracies, the effectiveness of the deep learning models was examined. The AUC represents the level or measure of separability, and the ROC (receiver operating characteristic) is a probability curve. The model performs better when predicting the intended classes when the AUC is larger. The FPR and TPR are displayed against each other on the ROC curve, with the FPR on the *x*-axis and the TPR on the *y*-axis. TPR (true positive rate)/recall/sensitivity is the proportion of actual positives that are accurately classified. The TPR is calculated using Equation (1):

$$TPR = TP/(TP + FN) \quad (1)$$

where TP = true positives; FN = false negatives; and FPR is the proportion of negative cases incorrectly identified as positive cases in the data. FPR is calculated using Equation (2):

$$FPR = FP/(FP + TN) \quad (2)$$

where FP = false positives; TN = true negatives; A confusion matrix can be defined as an M $\times$ M matrix that is used to assess the effectiveness of a classification model, where M defines the number of target classes. The matrix compares between the actual labels and the labels that are predicted by the neural network. A model's accuracy is calculated by dividing the number of classifications that are correctly predicted by the total number of predictions. The computational cost is essentially a measure of the amount of resources used by the neural network in training or inference, which is useful in determining how much time or computing power will be required to train or operate a NN. The number of operations required to execute a single instance of a particular model is expressed in floating-point operations per second (FLOPS). This will aid in describing the model's computational costs. The effectiveness of the training procedure for several deep learning models is examined based on the accuracy and loss. All these models have undergone training over 30 epochs, and the results are shown in Table 4. From Table 4, the loss and epoch graphs are almost inversely proportional to each other, which suggests that the models implemented for the mentioned dataset are learning towards convergence. The loss decreases from epoch 0 to epoch 10, and later it remains primarily constant. This explains that maximum model learning is achieved between epochs 0 and 10. "Accuracy vs. epochs" shows the same result as "loss vs. epochs", where the model increases its accuracy to the maximum between epochs 0 and 10. It can be observed that the validation accuracy is stagnant for the rest of the epochs, and to avoid overfitting the model, in this work the learning rate is reduced at 10^{-4} by a factor of 2 after two continuous epochs with no improvement in the validation accuracy.

Table 4. Performance analysis of the training process for deep learning models with real-time dataset collection.

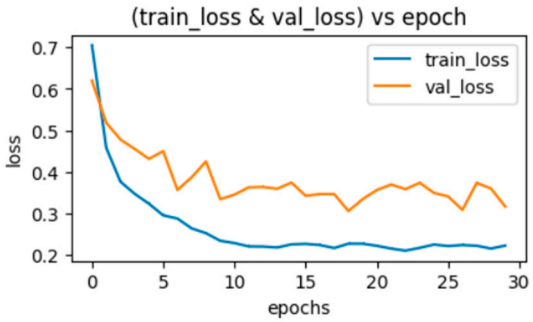
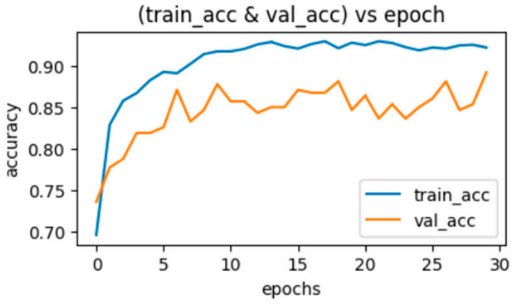
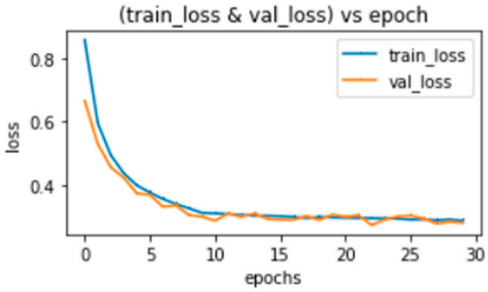
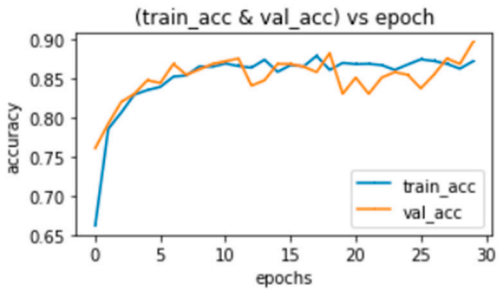
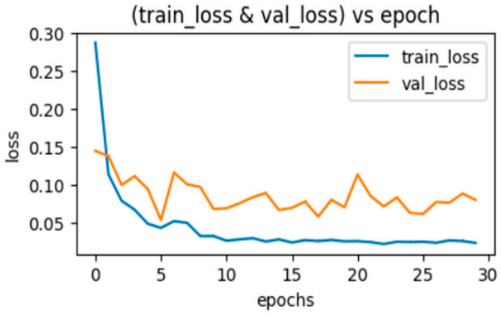
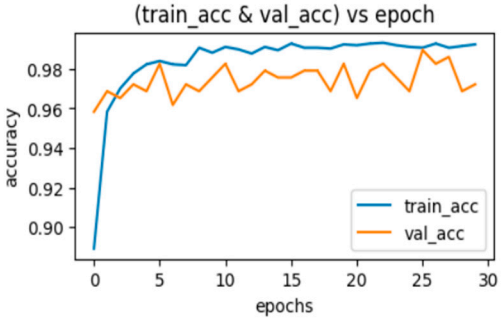
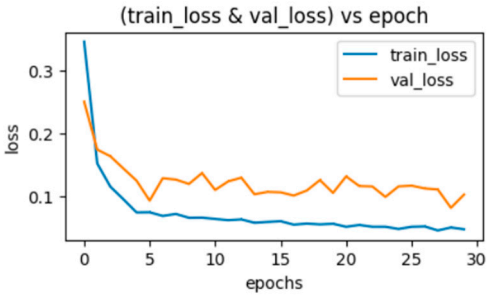
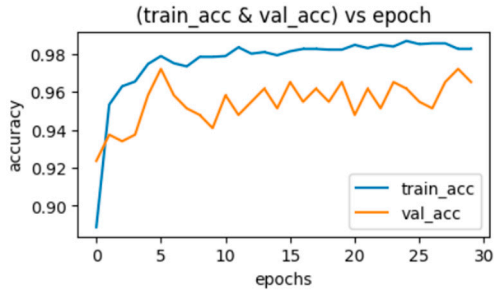
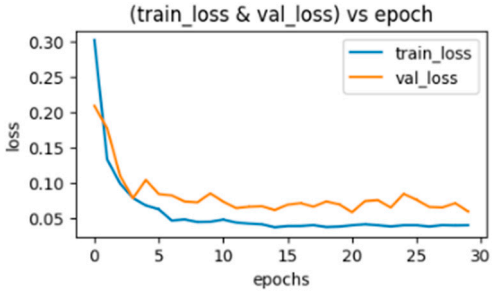
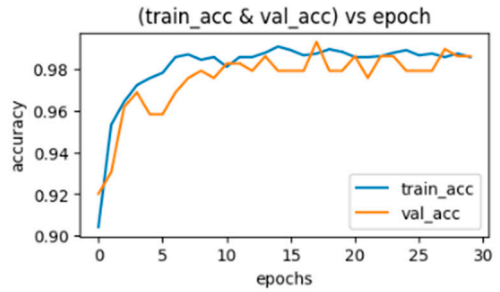
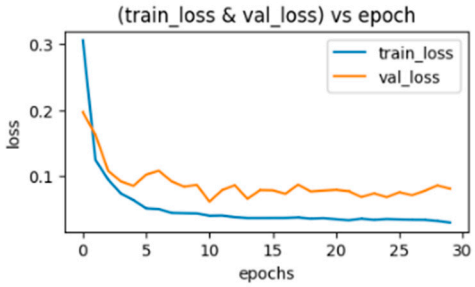
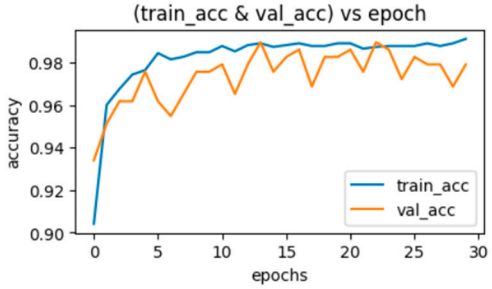
Model	Loss	Accuracy
MobileNet V2		
VGG19		
ResNet 50		
EfficientNet [B0]		
EfficentNet [B3]		

Table 4. *Cont.*

Model	Loss	Accuracy
EfficentNet [B4]		

Classification Results

The 3 × 3 confusion matrix for the best model (EfficientNetB4) is presented in Figure 7 with three rows and columns related to three different classes of vehicles. The rows represent the actual labels and columns represent the predicted labels. From the confusion matrix, it is understood that that the EfficientNetB4 model shows better performance in terms of handling non-over-volume goods carriers and non-goods carriers compared to over-volume goods carriers.

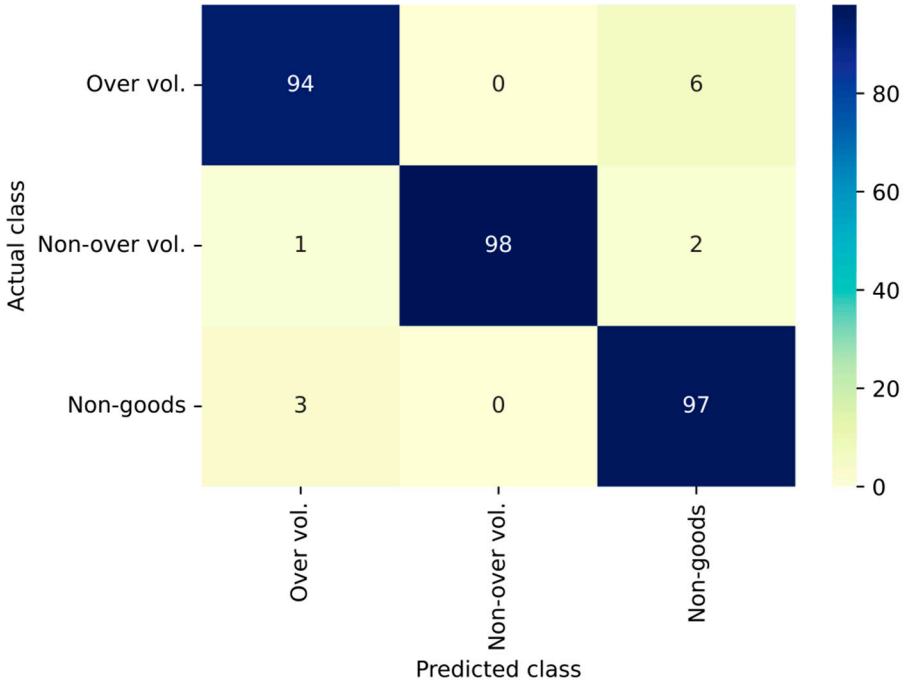


Figure 7. The 3 × 3 confusion matrix for the best model.

[EfficientNetB4]. Over Vol.—Over-volume goods carriers, Non-over vol.—Non-over-volume goods carriers, Non-goods—Non-goods carriers.

The size of the collected images is adjusted with the help of the image data generator class from the Keras library. The deep CNN models are trained on dimensions 224, 240, 300, and 380, and the best performance dimensions are presented in Table 5. From Table 5, it can be observed that the training performance of MobileNetV2 and VggNet were acceptable, but these models performed below par with the test data. In this work, the authors observed that the ResNet and EfficientNet architectures produced better performances for the training, validation, and testing sets. In detail, ResNet50 and EfficientNet [B0][B3][B4] performed similarly, with accuracies of 96%, 94%, 96%, and 96%, respectively. ResNet was

very effective with respect to accuracy, but it had high computational costs. EfficientNet architectures were effective with lower computational costs. Considering the FLOPS of each model, it is evident that EfficientNet [B0] was very effective with a trade-off between the computational costs of 0.7 billion FLOPS and the average accuracy of about 94%. The compound scaling method in EfficientNet architectures carefully balances network width, depth, and resolution to achieve better accuracy and efficiency. Powered by this compound scaling method, we demonstrated that EfficientNet scaled up very effectively, surpassing ResNet, VggNet, and MobileNet with an order of magnitude fewer parameters and FLOPs on our dataset.

Table 5. Analysis of performance measures for different deep learning models.

S. No.	Neural Network	Input Dimensions	Test Accuracy	Flop	Auc Score
1.	MobileNetv2	224 × 224	86.71	1.3 b	0.9001
2.	VGG 19	226 × 226	83.38	19.6 b	0.875
3.	ResNet 50	224 × 224	96.01	16.4 b	0.9701
4.	EfficientNet [B0]	240 × 240	94.01	0.7 b	0.9554
5.	EfficientNet [B3]	300 × 300	96.013	1.8 b	0.9700
6.	EfficientNet [B4]	380 × 380	96.34	4.2 b	0.9726

5. Conclusions

This work analyzes the performance of various deep CNN architectures for the classification of over-volume vehicles that can act as supporting blocks for implementing over-volume vehicle detection prototypes in real-time. The primary goal of this work is to evaluate the capabilities of cutting-edge deep convolutional neural networks for over-volume vehicle classification into non-over-volume goods carriers, over-volume-goods carriers, and non-goods carriers. A thorough analysis was performed with pre-trained models, namely MobileNetV2, VGGNet19, ResNet50, and EfficientNet[B0][B3][B4]. To the best of our knowledge, this is the first study that presents a quantitative analysis of the above-mentioned deep architectures for over-volume vehicle classification. The performance of these models was analyzed with metrics including the ROC curve, AUC score, confusion matrix, mean square error, and test accuracy. Observations with different architectures revealed that ResNet and EfficientNet architectures shows the best results in the required classification with Resnet50, EfficientNet [B3], and EfficientNet [B4] having an average accuracy of 95%. The future scope of this research may involve specificity in the data so that they are classified in terms of goods, as well as the addition of new classes that may help us to obtain a generic model. This study represents the first step in building a real-time prototype for over-volume vehicle detection.

Author Contributions: Conceptualization, V.D.M.V. and D.S.P.K.; methodology, V.D.M.V., P.V.S.N. and D.S.P.K.; software, V.D.M.V. and D.S.P.K.; validation P.V.S.N., S.S.R. and P.V.; formal analysis, V.D.M.V., S.S.R. and P.V.; investigation, V.D.M.V., P.V.S.N. and D.S.P.K.; data curation, V.D.M.V., P.V.S.N. and D.S.P.K.; writing—original draft preparation, V.D.M.V., P.V.S.N. and D.S.P.K.; writing—review and editing, S.S.R., P.V. and V.R.; visualization, S.S.R. and P.V.; supervision, S.S.R., P.V. and V.R.; project administration, S.S.R., P.V. and V.R. 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: Not applicable.

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

References

1. Adlinge, S.S.; Gupta, A.K. Pavement deterioration and its causes. *Int. J. Innov. Res. Dev.* **2013**, *2*, 437–450.
2. Shehu, Z.; Elma, N.; Endut, I.R.; Holt, G.D. Factors influencing road infrastructure damage in Malaysia. *Infrastruct. Asset Manag.* **2014**, *1*, 42–52. [CrossRef]
3. Rohim, M.Z.; Wijayanti, E.; Murti, A.C. Design of overloading detection systems on vehicles using adruino. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2021; Volume 1943.
4. Elwakil, E.; Eweda, A.; Zayed, T. Modelling the effect of various factors on the condition of pavement marking. *Struct. Infrastruct. Eng.* **2014**, *10*, 93–105. [CrossRef]
5. Copsey, N.; Drupsteen, L.; Kampen, J.V.; Kuijt-Evers, L.; Schmitz-Felten, E.; Verjans, M. *A Review of Accidents and Injuries to Road Transport Drivers*; European Agency for Safety and Health at Work: Luxembourg, 2010.
6. Bin Islam, M.; Kanitpong, K. Identification of factors in road accidents through in-depth accident analysis. *IATSS Res.* **2008**, *32*, 58–67. [CrossRef]
7. Hu, S.; Kong, M.; She, C. Design of vehicle overload detection system based on geophone. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2017; Volume 887, p. 012021.
8. Liu, Y.; Zhenhua, L. An Optimized Method for Dynamic Measurement of Truck Loading Capacity. In Proceedings of the 2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE), Singapore, 3–5 September 2018; IEEE: Piscataway, NJ, USA, 2018.
9. Singh, R.; Srivastava, G. Truck overloading detection and engine locking system. *Adv. Appl. Math. Sci.* **2022**, *21*, 7157–7162.
10. Kattimani, H.D.; Meghana, N.R.; Nagashree, B.; Sahana Munegowda, V.S. Vehicular Overload Detection and Protection. *Int. J. Latest Res. Eng. Technol. (IJLRET) ISSN*. 2017, pp. 119–122. Available online: http://www.ijlret.com/Papers/NC3PS2017/23.pdf (accessed on 17 December 2022).
11. Wang, Y.C.; Han, C.C.; Hsieh, C.T.; Fan, K.C. Vehicle type classification from surveillance videos on urban roads. In Proceedings of the 2014 7th International Conference on Ubi-Media Computing and Workshops, Ulaanbaatar, Mongolia, 12–14 July 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 266–270.
12. Zhang, F.; Xu, X.; Qiao, Y. Deep classification of vehicle makers and models: The effectiveness of pre-training and data enhancement. In Proceedings of the 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO), Zhuhai, China, 6–9 December 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 231–236.
13. Hicham, B.; Ahmed, A.; Mohammed, M. Vehicle Type Classification Using Convolutional Neural Network. In Proceedings of the 2018 IEEE 5th International Congress on Information Science and Technology (CiSt), Marrakech, Morocco, 21 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 313–316.
14. 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] [PubMed]
15. Rahman, M.; Rahman, S.; Ayoobkhan, M.U.A. Fine-Tuned Convolutional Neural Networks for Bangladeshi Vehicle Classification. In Proceedings of the 2022 International Conference on Innovations in Science, Engineering and Technology (ICISET), Chittagong, Bangladesh, 26–17 February 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 421–426.
16. 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; IEEE: Piscataway, NJ, USA, 2009; pp. 248–255.
17. Tabassum, S.; Ullah, S.; Al-Nur, N.H.; Shatabda, S. Poribohon-BD: Bangladeshi local vehicle image dataset with annotation for classification. *Data Brief* **2020**, *33*, 106465. [CrossRef] [PubMed]
18. 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]
19. Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Learning and transferring mid-level image representations using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1717–1724.
20. Razavian, A.S.; Azizpour, H.; Sullivan, J.; Carlsson, S. Cnn features off-the-shelf: An astounding baseline for recognition. *arXiv* **2014**, arXiv:1403.6382.
21. Kieffer, B.; Babaie, M.; Kalra, S.; Tizhoosh, H.R. Convolutional neural networks for histopathology image classification: Training vs. Using pre-trained networks. In Proceedings of the 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), Montreal, QC, Canada, 28 November–1 December 2017; pp. 1–6.
22. Li, X.; Pang, T.; Xiong, B.; Liu, W.; Liang, P.; Wang, T. Convolutional neural networks based transfer learning for diabetic retinopathy fundus image classification. In Proceedings of the 2017 10th International Congress on Image and Signal Processing, Biomedical Engineering and Informatics (CISP-BMEI), Shanghai, China, 14–16 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–11.
23. Carneiro, G.; Nascimento, J.; Bradley, A.P. Unregistered multiview mammogram analysis with pre-trained deep learning models. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2015; pp. 652–660.

24. Gao, M.; Bagci, U.; Lu, L.; Wu, A.; Buty, M.; Shin, H.-C.; Roth, H.; Papadakis, G.Z.; Depeursinge, A.; Summers, R.M.; et al. Holistic classification of ct attenuation patterns for interstitial lung diseases via deep convolutional neural networks. *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.* **2015**, *6*, 1–6. [CrossRef] [PubMed]
25. Shin, H.-C.; Roth, H.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging* **2016**, *35*, 1285–1298. [CrossRef] [PubMed]
26. Tajbakhsh, N.; Shin, J.; Gurudu, S.; Hurst, T.; Kendall, C.; Gotway, M.; Liang, J. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Trans. Med. Imaging* **2016**, *35*, 1299–1312. [CrossRef] [PubMed]
27. Azizpour, H.; Razavian, A.S.; Sullivan, J.; Maki, A.; Carlsson, S. From generic to specific deep representations for visual recognition. In Proceedings of the Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–12 June 2015; pp. 36–45.
28. Penatti, O.A.B.; Nogueira, K.; Dos Santos, J.A. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In *Computer Vision and Pattern Recognition Workshops*; IEEE: Piscataway, NJ, USA, 2015; pp. 44–51.
29. Dong, K.; Zhou, C.; Ruan, Y.; Li, Y. MobileNetV2 Model for Image Classification. In Proceedings of the 2020 2nd International Conference on Information Technology and Computer Application (ITCA), Guangzhou, China, 18–20 December 2020; pp. 476–480. [CrossRef]
30. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
31. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018), Salt Lake City, UT, USA, 18–23 June 2018.
32. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning Proceedings of Machine Learning Research 97, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114. Available online: https://proceedings.mlr.press/v97/tan19a.html (accessed on 12 August 2022).
33. 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]

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.