RESEARCH

A new approach based on convolutional neural network and feature selection for recognizing vehicle types

Gürkan Doğan[1] · **Burhan Ergen**[2]

Abstract

The number of vehicles used in traffic life has reached enormous dimensions today. The increase in the number of vehicles day by day causes some traffic problems along with it; such as traffic congestion, accidents, pollution, and safety. To overcome all these problems, convolutional neural networks (CNN) methods are one of the trend methods used in recent years due to their success. In this study, a new approach is proposed to use this power of CNN in low-power devices. First of all, MobileNetv1, MobileNetv2, and NASNetMobile models were optimized to increase accuracy performance. Then, an approach is proposed in which these optimized mobile CNN approaches are used only as feature extractors, and methods such as combining, selecting, and classifying the obtained features are used together. As a result of the classification made with this approach, the classification accuracy has increased by approximately 5%.

Keywords Deep learning · Feature selection · Classification · Vehicle types

1 Introduction

In recent years, urban traffic congestion and its associated impacts remain a major problem in big cities around the world. Due to the rapid increase in the number of vehicles registered to traffic, city managers have put forward alternative heavy vehicle restriction strategies to reduce the negative impact of heavy tonnage vehicles on traffic. One of the most common strategies is restrictions on vehicles of certain weights from using designated lanes and routes. Time restriction is another type of restriction strategy that typically restricts the movement of heavy vehicles on a particular road during times of increased traffic congestion. The main purpose of implementing such restriction strategies is to minimize the time in transport, minimize the cost of road maintenance, reduce carbon emission in certain regions, reduce fuel consumption, and maximize pedestrian

and driver safety [1–3]. Similar restriction strategies are applied in big cities such as London and Melbourne. Certain vehicle types may be restricted from joining in traffic for all restriction strategies to succeed. Therefore, it is important to recognize the types of vehicles in traffic with high success.

On the other hand, deep learning-based methods are trending technologies used to recognize vehicle types [4–6]. While traditional methods are successful only in specific tasks, methods based on deep learning are also successful in different tasks. Also, deep learning-based methods extract features automatically, while traditional methods extract them manually. For these reasons, methods based on deep learning are preferred more than traditional methods. In particular, the increasing use of mobile devices in daily life and the successful application of deep learning techniques in various fields encourage researchers to use deep learning methods on mobile devices [7]. Deep learning methods such as MobileNet [8] (aka MobileNetv1), MobileNetv2 [9], and NASNetMobile [10] are also called mobile deep learning methods because they can be used on devices that require low resource consumption [11]. Such mobile CNN models in the literature do not provide sufficient success to recognize vehicle types when used alone. In particular, if the data set consists of images obtained from the real environment, it becomes more difficult to determine the type of vehicles since there are many objects of various dimensions in an image.

✉ Gürkan Doğan
  gurkandogan@munzur.edu.tr

  Burhan Ergen
  bergen@firat.edu.tr

1 Department of Computer Engineering, Faculty of Engineering, Munzur University, Tunceli, Turkey

2 Department of Computer Engineering, Faculty of Engineering, Fırat University, Elazig, Turkey

Moreover, the greater the number of vehicle types in the data set, that is, the number of classes, the more difficult it is to identify the vehicle type. In this study, a new approach is proposed to overcome such difficult problems. For the training and testing of the proposed approach, a data set consisting of 3972 real images collected from the real environment and containing seven different vehicle types was used. Recognizing vehicle types is a difficult task as the dataset used contains complex traffic scenes and has a multi-class structure.

In this study, a comprehensive study was carried out for vehicle images obtained from a real environment, and a new approach based on mobile convolutional neural networks, feature selection, and classification is proposed. The proposed study consists of two main stages. In the first stage, MobileNetv1, MobileNetv2, and NASNetMobile CNN models, which can be used on mobile devices, were selected for the recognition of vehicle types. Then, for these mobile CNN models to perform the best learning, while the top block layers were removed, it was optimized by adding layers such as global average pooling, dropout, and fully connected, as well as transfer learning. The optimized versions of the MobileNetv1, MobileNetv2, and NASNetMobile models are named OMobileNetv1, OMobileNetv2, and ONASNet-Mobile, respectively. In the second stage, an approach is proposed in which the optimized mobile CNN models are used only as feature extractors, and methods such as combining, selecting, and classifying extracted features are used together. Optimized mobile CNN models are used for feature extraction only. The layers used to extract features in OMobileNetv1, OMobileNetv2, and ONASNetMobile models are 'dropout', 'dropout', and 'avg_pool' layers, respectively. By combining different numbers of features obtained from these layers, 3360 new features were created. At this point, size reduction was not made for the features. Because it was necessary to select the best features from the entire set of features. Then, a new feature set was obtained from these features using the Linear Support Vector Classification (SVC) method. Finally, for this new feature set, a high success rate of 87.7% was achieved by using the SVM method with known classification success. It is seen that the new features obtained as a result of this study greatly increase the classification success. The contributions of this study to the literature are as follows;

- Recognizing seven different types of vehicles in a dataset of images obtained from the real environment with high success,
- Using three different convolutional neural networks as feature extractors and creating a new feature set from the extracted features,
- To ensure that the best features are selected by using the feature selection algorithm,

- The proposed approach is to compare and analyze other state-of-the-art methods.

The remainder of this article is organized as follows; while the related works are given in the second chapter, the data set, the mobile CNN models, and some feature selection and classification methods are given in the third chapter. In the fourth chapter, the proposed approach including structures such as feature concatenating, selection, and classification is presented, while in the fifth chapter the experimental strategy, analysis, and results are presented. Finally, conclusions and future works are shared in the sixth chapter.

2 Related works

Traditional methods were used in most of the studies for vehicle classification. In these methods, solutions are proposed only for certain tasks by extracting the background and other features from the vehicle image using different algorithms. Lee et al. [12], with the HOG and SVM techniques, Manzoor and Morgan [13], used the Scale-invariant feature transform (SIFT) technique to feature extraction and proposed an SVM-based classification model. Seenouvong et al. [14], proposed a classification model based on a virtual detection line (VDL) and a virtual detection zone (VDZ). The features obtained from VDL and VDZ are classified with KNN. Zhang [15], combined PHOG and Gabor Transform feature identification methods and proposed the cascade classifier scheme to recognize the vehicle type. Psyllos [16] proposed a model for vehicle manufacturers and model recognition (VMMR). While SIFT is used to extract features, a probabilistic neural network (PNN) is used for classification.

Unlike traditional methods that use manual feature extraction, Convolutional Neural Network (CNN) provides generalized solutions for multiple tasks, not just for the solution of a specific task, and performs feature extraction automatically [17]. Rabano et al. [18] used the MobileNetv1 model for the classification of wastes. This model classified glass, paper, cardboard, plastic, metal, and other trash with 87.2% accuracy. Bi et al. [19] proposed a MobileNetv1-based method to identify apple leaf diseases. This method performs the task of binary classification as Alternaria leaf blotch and rust leaf and reaches 73.5% accuracy. In [20] and [21], MobileNetv2, NASNetMobile, and other deep learning methods were used to classify Covid-19 disease. Mishra et al. [4] proposed the Modified Sine Cosine Algorithm–Particle Swarm Optimization (MSCA–PSO) based MobileNetv2 Convolutional Neural Network (CNN) model to classify diseases of avocado fruit as healthy and unhealthy from images. Their proposed method performed a binary classification task with approximately 98.4% accuracy. Wijaya et al. [22] proposed a CNN-based method for the classification of citrus fruits.

In this study, grapefruit and citrus separation was achieved with an accuracy of approximately 96% by using features such as diameter size, weight and average RGB value of citrus fruits and using a two-class study. On the other hand, in studies to classify vehicle types, Luo et al. [6] proposed a new neural network-based classifier method called Dense-TNT to classify sedan and pickup vehicle types obtained from satellite images. This method achieved an accuracy success of approximately 79.4% in images obtained in light foggy weather conditions. Swastika et al. [23] used the adadelta optimizer and the MiniVGGNet CNN model to classify the three types of vehicles, achieving the highest success with 73% accuracy. Similar to the method that uses the deep learning method for feature extraction and classifies real vehicle images with these features, there is not a common study in the literature. However, similar studies have been successfully applied in other fields, such as the flower classification in [24].

While the traditional methods used to classify the types of vehicles were successful only in specific tasks, deep learning methods were more successful in different tasks than traditional methods. However, distinguishing vehicles from each other is a challenging task in deep learning methods, as images collected from the real environment contain complex scenes. Although the studies in the literature generally distinguish between two or three vehicle types, their accuracy success has reached approximately 80%. In addition, there are dozens of vehicle types in a real traffic scene, and the similarity rate increases as the number of vehicle types increases. Besides, the similarity of the color, size, and amount of illumination of the vehicles is another factor that complicates the classification task. In this study, a new approach is proposed to overcome these difficult problems. The proposed approach aims to automatically recognize seven different vehicle types with a high success rate using a dataset containing images from real traffic scenes.

3 Dataset and methods

3.1 Vehicle dataset

In this study, a vehicle data set containing different vehicle types collected from real traffic scenes by Sandeep was used [25]. The vehicle data set consists of approximately 28,000 vehicle images and 17 different categories obtained from the real environment at different times and angles. The actual data set was obtained with 3972 vehicle images, 7 different categories, and a 224 × 224 frame size by reducing category and frame size, the number of images of this data set. The new data set was obtained by randomly selecting the vehicle images. The vehicle data set of which several sample images are given in Fig. 1, includes images of 88 ambulances, 646 bicycles, 682 buses, 690 cars, 527 taxis, 624 trucks, and 715 vans. Later, 80% of the data set was allocated for training, 10% for validation, and 10% for testing.

3.2 Mobile CNN models

MobileNetv1 is an efficient neural network consisting of a deeply separable convolution layer using a depth and 1 × 1 convolution layer. Based on the theory that smaller models have fewer memorization problems, MobileNetv1 mostly uses it in depth as it has less complexity [8, 9]. On the other hand, MobileNetv2, which is an upper version of MobileNetv1, implements the new layer block, which is two types of Bottleneck layers that differ according to the stride. In MobileNetv2, instead of the Bottleneck layer consisting only of a convolutional layer, an in-depth convolution layer is used, which aims to reduce this parameter further [9]. NAS-NetMobile performs better learning by using the learning gained from training smaller data sets with the training of a larger data set. For example, NASNetMobile can be used

Fig. 1 A sample image from the vehicle dataset

to train first on CIFAR10 and then perform training on ImageNet [10].

In this study, mobile CNN models were preferred because of their compact structure, resource consumption, high accuracy, and speed. Besides, network architectures with different structures were preferred because they were thought to be strong in distinguishing different features of an image [26].

3.3 SVM classifier

Support vector machine (SVM) is a statistical machine learning method that uses the hyperplane that maximizes the margin in a given training data to find a class boundary and separate two classes [27]. In this method, a hyperplane with the best aperture for two different classes is sought. On the other hand, the basic SVM method has been generalized to separate data with two or more classes [28]. In this way, SVM has become a powerful tool for nonlinear classification, regression, and multivariate function prediction. Radius based function (RBF), one of the most popular kernel types used in SVM, can be applied to any distribution of sampling with the selection of parameters. It is increasingly used in nonlinear applications of support vector machines [29, 30]. In this study, SVM and RBF combinations are used to classify vehicle images.

3.4 Feature selection methods

Feature selection is the process of extracting a new subset of features from existing features in the data set. With this process, unnecessary and unrelated features are removed. In this way, both the size of the data is reduced and the performance is increased [31]. The feature selection process is achieved by using feature selection methods. Feature selection methods aim to increase the success of classification by selecting the best features in a data set. Feature selection methods are divided into three categories: filter, wrapper, and embedded methods. In the filter method, the features directly rank according to different performance evaluation criteria and select the first N features with the highest score. In wrapper methods, a subset of features is selected when a model is trained through classification algorithms. Some common wrapper methods are forward selection, backward elimination, and recursive feature elimination. Embedded methods are similar to wrapper methods. In these methods, feature selection is linked to classification algorithms. The embedded methods are a combination of filter and wrapper methods. It uses classification algorithms with built-in capabilities to select features [32, 33]. In this study, feature selection was made using embedded feature selection methods such as Ridge Cross-Validation (RidgeCV), Stochastic Gradient Descent Classifier (SDGC), and Linear Support Vector Classifier (LSVC). Feature selection cannot be done directly with

these methods. However, features can be selected from the model by using the feature selection technique. This technique is a meta-transformer that can be used with an estimator after training to determine the significance of each feature [34]. If the significance corresponding to the feature values is below the provided threshold parameter, the features are considered insignificant and removed. In addition to specifying the threshold numerically, there are heuristic methods of finding a threshold using a set of parameters. Available heuristic methods are mean, median, and multiples of the mean obtained by multiplying the decimal constant numbers.

4 Proposed approach

The proposed approach consists of two main stages. In the first stage, mobile CNN models such as MobileNetv1, MobileNetv2, and NASNetMobile were selected due to their compact structure, low resource consumption, high accuracy, and speed. In addition, three different network models were preferred since network architectures with different structures are thought to be strong in distinguishing different attributes of an image. Then, for these mobile CNN models to perform the best learning, while the top block layers were removed, it was optimized by adding layers such as global average pooling, dropout, and fully connected, as well as transfer learning. The optimized versions of the MobileNetv1, MobileNetv2, and NASNetMobile models are named OMobileNetv1, OMobileNetv2, and ONASNetMobile, respectively. Optimized mobile CNN models are presented in Fig. 2. The optimization of these models consists of seven steps. The three models get the vehicle data set as input in the first step and use the pre-trained ImageNet data set with the transfer learning method in the second step. In the third step, all convolution layers in the original MobileNetv1, MobileNetv2, and NASNetMobile architectures are used in the OMobileNetv1, OMobileNetv2, and ONASNetMobile approach, respectively. In the fourth step, the weights of all layers pre-trained in the original MobileNetv1, MobileNetv2, and NASNetMobile architectures are frozen in the OMobileNetv1, OMobileNetv2, and ONASNetMobile approach, respectively. If all the layers are trained at this point, the updates to the greatness of the gradient will be very large due to random weights from the classifier, and the pre-trained model will forget what it has learned. Therefore, updating the weights of the pre-trained model during training is avoided. In the fifth step, the global average pooling (GAP) layer is added for the three approaches. In the sixth step, the dropout layer was added to the OMobileNetv1 and OMobileNetv2 approaches, and the full connected (FC) layer to the ONASNetMobile approach. Global average pooling (GAP) and dropout layers have been added to further reduce model parameters and prevent memorization. In the

Fig. 2 Sematic representation of optimized convolutional neural networks used in the proposed approach. **a** OMobileNetv1, **b** OMobileNetv2 and, **c** ONASNetMobile


(a)   (b)   (c)
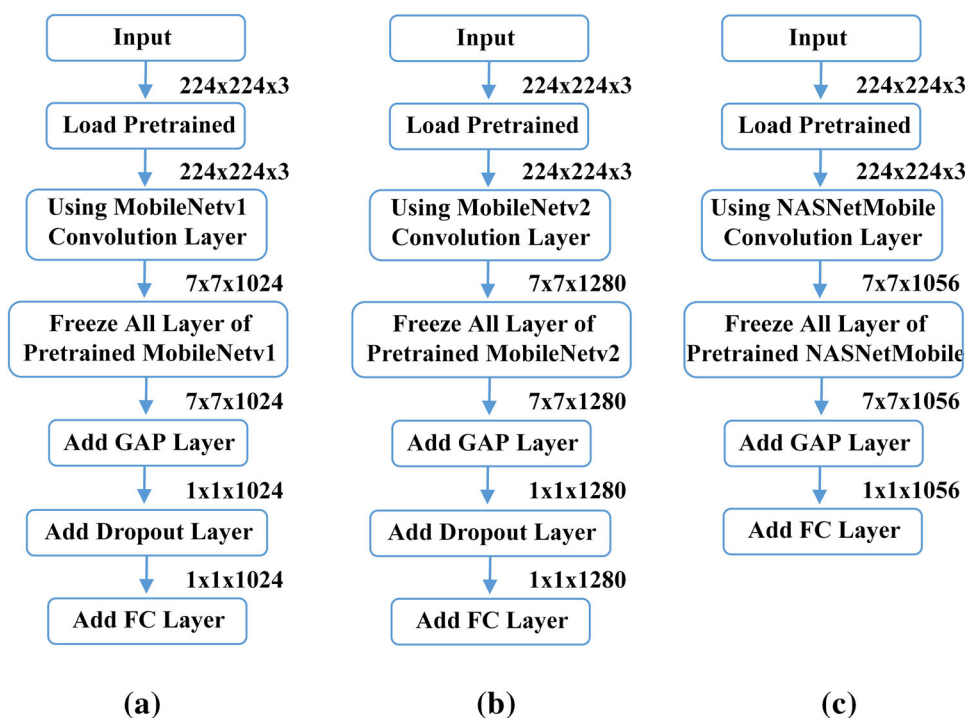
last layer, a fully connected layer was added for the OMobileNetv1 and OMobileNetv2 approaches. In addition, the softmax activation function was added to the last layers of these models to measure classification success. In the second stage, optimized mobile CNN models were used only as feature extractors. The layers used to extract features in OMobileNetv1, OMobileNetv2, and ONASNetMobile models are 'dropout', 'dropout', and 'avg_pool' layers, respectively. By combining different numbers of features obtained from these layers, 3360 new features were created. At this point, size reduction was not made for the features. At this point, no dimension reduction was made for the extracted features from each optimized mobile CNN model. Because it was necessary to select the best features from the entire set of features. However, after the features were combined, a new feature set was obtained from these features using the Linear Support Vector Classification (SVC) method. In this way, it was ensured that the best features were selected from the features extracted from three different network models. Then, these new features were classified by the SVM method with known classification success and final results were obtained. Figure 3 shows the schematic of the proposed approach.

5 Experiments

5.1 Experimental strategy

In this study, the batch size is set to 32 for training the mobile CNN models such as ONASNetMobile, OMobileNetv1, and OMobileNetv2. The dropout rate for the OMobileNetv1 and OMobileNetv2 methods was set to 0.065 and 0.09, respectively. The Adam [35] optimization algorithm, which gives the best results, was used in all three models. Different learning rates for the three models were tested iteratively and the optimal learning rate was determined as 1×10^{-5}. While the Softmax function is used as the activation function of the output layer in mobile CNN models, categorical cross entropy is used as the loss function. Three different feature selection methods, namely RidgeCV, SDGC, and LSVC, were used for experimental tests. For one of these methods, Ridge Cross Validation (RidgeCV), Sklearn's default parameters are used. For the other methods, the Stochastic Gradient Descent Classifier (SDGC), the stop criterion parameter is set to 1×10^{-4}. The regularization parameter for the Linear Support Vector Classifier (LSVC) is selected as 1×10^{-2}. The norm parameter used in the penalty is L1 (Lasso). Since the number of samples in the data set is greater than the number of features, the dual parameter is selected as false and the stop criterion parameter is selected as 1×10^{-5}. Finally, the Support Vector Classifier is used to classify the obtained features and the kernel type of this classifier was selected as RBF.

Fig. 3 Schematic of the proposed approach
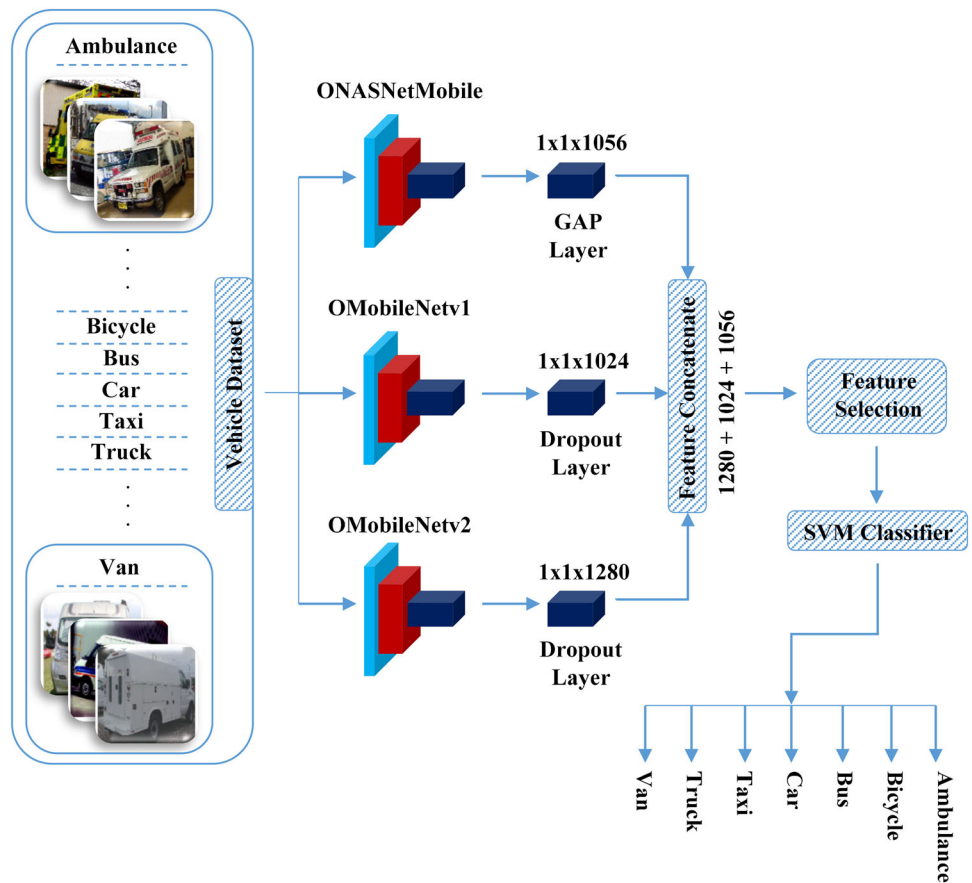
5.2 Experimental analysis and results

In this study, three different classifications were made with optimized mobile CNN models such as ONASNetMobile (ONN), OMobileNetv1 (OMN1), and OMobileNetv2 (OMN2). In this way, it will be possible to compare the proposed approach and mobile CNN models in terms of accuracy success. Then, the features extracted from the relevant layers of these mobile CNN models are combined and three different sub-feature sets are created by choosing from the combined features using the feature selection methods. Then, these subsets of features are individually classified by the SVM. In this way, the contribution of each feature selection method to the proposed approach can be determined. To evaluate the results, the criteria obtained from the complexity matrix are examined; precision (Prec.), recall (Rec.), f-score (F-Scr.), and accuracy (Acc.). The values of these metrics are found by the following equations. In these equations, TP denotes to true-positive, TN, true-negative, FP, false-positive, FN, and false-negative [36].

$$\text{Prec} = \text{TP}/(\text{TP} + \text{FP}) \tag{1}$$

$$\text{Rec} = \text{TP}/(\text{TP} + \text{FN}) \tag{2}$$

$$F-\text{Scr} = 2\text{TP}/(2\text{TP} + \text{FP} + \text{FN}) \tag{3}$$

$$\text{Acc} = (\text{TP} + \text{TN})/(\text{TP} + \text{FN} + \text{TN} + \text{FP}) \tag{4}$$

The experimental study is generally carried out in three different stages. In the first stage, transfer learning is carried out with the three different optimized mobile CNN models, classification is made separately and the results are evaluated. In the second stage, binary combinations of three separate feature sets obtained from these three optimized mobile CNN models are selected and combined. Sub-feature sets are then created from the combined features using feature selection methods. The sub-feature sets are separately classified by SVM and the results are evaluated. In the last stage, the triple combination of three separate feature sets obtained from the three optimized mobile CNN models is selected and combined. The sub-feature sets are then created from the combined features using feature selection methods. Finally, the sub-feature sets are separately classified by SVM and the results are evaluated.

When the vehicle dataset is trained with optimized mobile CNN models, the accuracy graph of the training and validation sets is as in Fig. 4. The best training and validation harmony in these graphics was achieved in the ONASNet-Mobile model. On the other hand, the highest accuracy rate
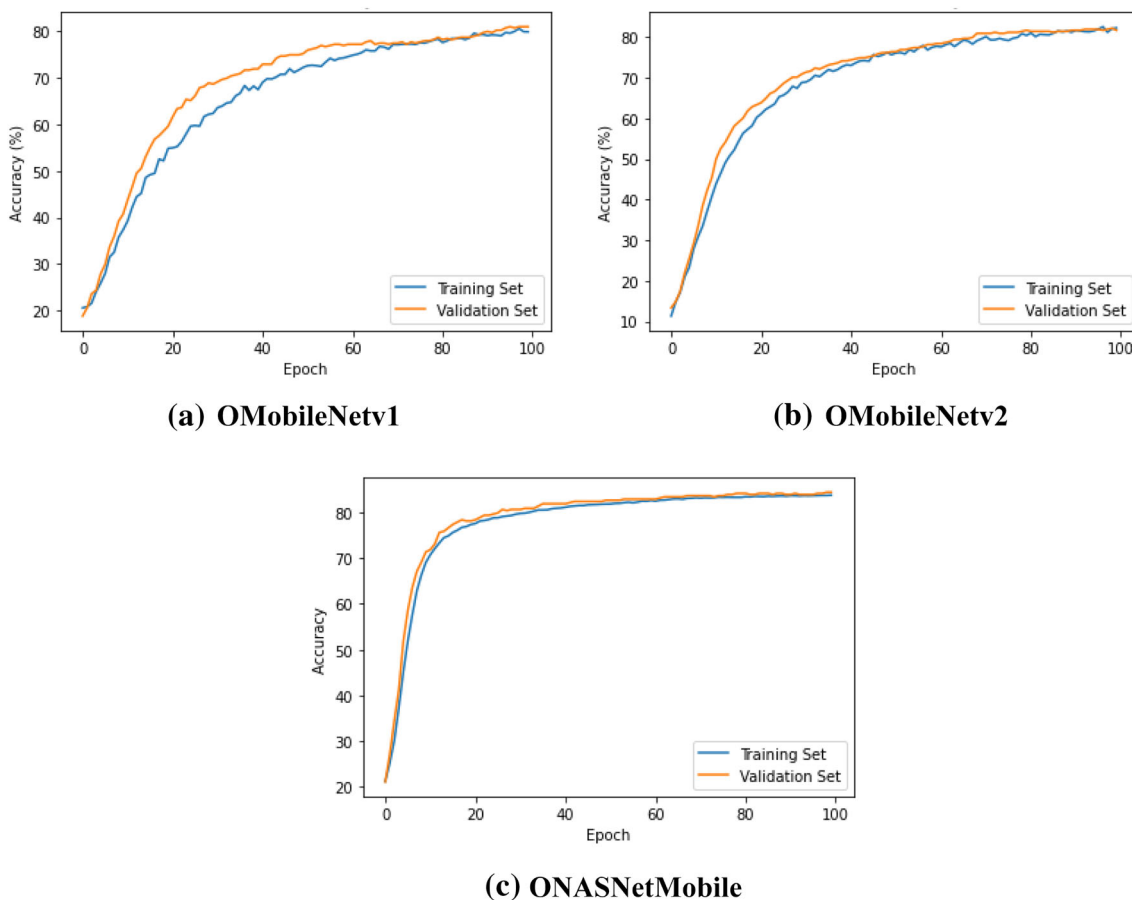
(a) OMobileNetv1

(b) OMobileNetv2

(c) ONASNetMobile

Fig. 4 Accuracy of training and validation sets for proposed methods
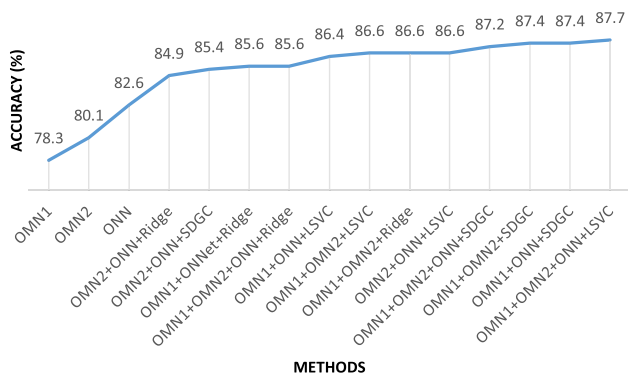

Fig. 5 Accuracy rates according to the methods used in the proposed approach

in the classification was achieved in this model at 82.6%. As seen in Fig. 5 and Table 1, the worst accuracy rate among these CNN models was realized in OMobileNetv1 with 78.3%. Among the features combined as a binary combination, the highest accuracy rate was obtained from the feature set consisting of the combination of the binaries such as OMobileNetv1 (OMN1), OMobileNetv2 (OMN2),

and OMobileNetv1 (OMN1), ONASNetMobile (ONN). The best feature selection from the feature set created from these binary combinations was obtained by the Stochastic Gradient Descent classifier method. As a result, the highest classification success rate of 87.4% was achieved with the properties selected from these binary combinations. The worst classification success was obtained from binary feature combinations, the combination of OMobileNetv2 and ONASNetMobile CNN models, while the worst performance in feature selection was achieved with RidgeCV.

When triple feature selection combinations are examined, the highest classification success is obtained when features are selected with the linear support vector classifier method. The lowest classification success is achieved when the feature selection is made with the RidgeCV method.

Finally, as a result of the experimental tests, the highest accuracy success was obtained with 87.7% by combining the features extracted from the OMobileNetv1, OMobileNetv2 and ONASNetMobile CNN methods and by selecting the features with the LSVC method and classifying with the SVM method. Therefore, in this study, a combination of OMobileNetv1, OMobileNetv2, ONASNetMobile CNN methods

Table 1 Classification results of proposed approaches with SVM

Methods	Metrics	Ambulance	Bicycle	Bus	Car	Taxi	Truck	Van	Accuracy
OMobileNetv1	Precision	0.50	0.91	0.93	0.65	0.85	0.71	0.66	78.3
	Recall	0.08	0.97	0.96	0.72	0.80	0.69	0.68	
	F1-Score	0.14	0.94	0.94	0.68	0.83	0.70	0.67	
OMobileNetv2	Precision	0.83	0.97	0.89	0.69	0.79	0.75	0.70	80.1
	Recall	0.42	1.00	0.96	0.74	0.77	0.69	0.69	
	F1-Score	0.56	0.98	0.92	0.71	0.78	0.72	0.70	
ONASNetMobile	Precision	0.78	0.89	0.89	0.70	0.84	0.85	0.83	82.6
	Recall	0.58	0.95	1.00	0.81	0.74	0.77	0.71	
	F1-Score	0.67	0.92	0.94	0.75	0.79	0.81	0.77	
OMobileNetv1 + OMobileNetv2 + LSVC	Precision	0.75	0.99	0.92	0.77	0.82	0.93	0.81	86.6
	Recall	0.43	0.93	0.97	0.86	0.74	0.88	0.85	
	F1-Score	0.55	0.96	0.94	0.81	0.78	0.90	0.83	
OMobileNetv1 + OMobileNetv2 + Ridge	Precision	0.60	0.99	0.91	0.80	0.79	0.93	0.81	86.6
	Recall	0.43	0.95	0.90	0.88	0.76	0.88	0.85	
	F1-Score	0.50	0.97	0.90	0.84	0.78	0.90	0.83	
OMobileNetv1 + OMobileNetv2 + SDGC	Precision	0.75	0.99	0.92	0.77	0.89	0.94	0.81	87.4
	Recall	0.43	0.93	0.95	0.90	0.78	0.86	0.85	
	F1-Score	0.55	0.96	0.93	0.83	0.83	0.90	0.83	
OMobileNetv1 + ONASNetMobile + LSVC	Precision	0.60	0.99	0.95	0.80	0.83	0.87	0.79	86.4
	Recall	0.43	0.93	0.90	0.88	0.80	0.84	0.85	
	F1-Score	0.50	0.96	0.92	0.84	0.82	0.86	0.82	
OMobileNetv1 + ONASNetMobile + Ridge	Precision	0.50	0.99	0.96	0.80	0,80	0.89	0.77	85.6
	Recall	0.43	0.93	0.86	0.88	0.78	0.86	0.83	
	F1-Score	0.46	0.96	0.91	0.84	0.79	0.88	0.80	
OMobileNetv1 + ONASNetMobile + SDGC	Precision	0.67	0.99	0.98	0.80	0.85	0.92	0.78	87.4
	Recall	0.57	0.93	0.91	0.90	0.78	0.86	0.86	
	F1-Score	0.62	0.96	0.95	0.84	0.81	0.89	0.82	
OMobileNetv2 + ONASNetMobile + LSVC	Precision	0.50	0.99	0.91	0.81	0.87	0.89	0.80	86.6
	Recall	0.43	0.95	0.86	0.91	0.78	0.86	0.85	
	F1-Score	0.46	0.97	0.88	0.86	0.82	0.88	0.82	
OMobileNetv2 + ONASNetMobile + Ridge	Precision	0.50	0.97	0.92	0.78	0.88	0.86	0.77	84.9
	Recall	0.43	0.93	0.83	0.88	0.76	0.86	0.83	
	F1-Score	0.46	0.95	0.87	0.83	0.82	0.86	0.80	
OMobileNetv2 + ONASNetMobile + SDGC	Precision	0.50	0.97	0.93	0.76	0.86	0.89	0.79	85.4
	Recall	0.43	0.95	0.88	0.90	0.76	0.86	0.79	
	F1-Score	0.46	0.96	0.90	0.82	0.81	0.88	0.79	
OMobileNetv1 + OMobileNetv2 + ONASNetMobile + LSVC	Precision	0.60	0.99	0.96	0.80	0.85	0.91	0.80	**87.7**
	Recall	0.43	0.93	0.93	0.90	0.80	0.88	0.85	
	F1-Score	0.50	0.96	0.95	0.85	0.82	0.89	0.82	
OMobileNetv1 + OMobileNetv2 + ONASNetMobile + Ridge	Precision	0.40	0.99	0.98	0.78	0.84	0.86	0.77	85.6
	Recall	0.29	0.93	0.86	0.88	0.76	0.89	0.83	
	F1-Score	0.33	0.96	0.92	0.83	0.80	0.88	0.80	
OMobileNetv1 + OMobileNetv2 + ONASNetMobile + SDGC	Precision	0.50	0.99	0.93	0.80	0.85	0.94	0.80	87.2
	Recall	0.43	0.93	0.93	0.91	0.80	0.86	0.82	

Table 1 (continued)

Methods	Metrics	Ambulance	Bicycle	Bus	Car	Taxi	Truck	Van	Accuracy
	*F*1-Score	0.46	0.96	0.93	0.85	0.82	0.90	0.81	

Table 2 Comparison of the proposed approach with state-of-the-art methods

Methods	No. of classes	No. of samples	Dataset	Accuracy (%)
SS-BLS [37]	6	2700	BIT-Vehicle	71.33
MiniVGGNet [23]	3	2930	Private	73.00
Dense-TNT (s24) [6]	2	57,984	COWC	79.41
VGG19 [38]	11	786,702	MIO-TCD	83.00
Proposed Approach	7	3972	Vehicle Dataset	87.70

for feature extraction, LSVC method for feature selection and SVM method for classification is proposed.

The comparison of the proposed approach with other state-of-the-art methods in the literature is given in Table 2. In this comparison table, the number of samples and the number of classes in the relevant data set are also considered important factors in order to evaluate the classification success of a method. The SS-BLS method in this table is a new flat network model developed by Chen et al. [39] by inheriting most of the features and superiorities of the random vector functional-link neural networks (RVFLNN). This model achieved an accuracy of 71.33% when trained and tested with the BIT Vehicle dataset by Guo et al. [37]. Although the number of samples and the number of classes in the data set in which this method is tested is close to that of our proposed approach, the accuracy of our proposed approach is 16% higher. VGG19 [38] and Dense-TNT [6], one of the state-of-the-art methods based on other convolutional neural networks, have a lower accuracy success than our proposed approach, despite using datasets containing a very large number of samples. Our proposed approach achieved approximately 5% higher accuracy than its closest competitor, the VGG19.

6 Conclusions and future works

In this study, we propose a new approach that achieves a higher classification success than state-of-the-art methods using the vehicle dataset of real-world images. The proposed approach is developed based on mobile CNN models such as MobileNetv1, MobileNetv2, and NASNetMobile to recognize seven different types of vehicles. In the proposed approach, the reason for using these mobile CNN models is because they are lightweight, fast speed, and consume fewer hardware resources. However, these models do not provide sufficient accuracy for vehicle type recognition when used alone. This is because the dataset contains images from the

real environment, there are seven different types of vehicles, and the images contain complex traffic scenes. All these situations are problems that make it difficult to recognize vehicle types. To overcome these problems, first of all, three different mobile CNN models were optimized. Then, features were extracted and combined with optimized CNN models. Using the Linear Support Vector Classification (LSVC) feature selection method, a new feature set is created from the combined features. In this way, it is ensured that the best features are selected. Then, using the Support Vector Machine, whose success in classification is known, 87.7% accuracy was achieved. This result is approximately 5% higher than the highest classification success of state-of-the-art methods. In our future studies, we aim to develop the proposed approaches with different datasets and different feature selection methods to achieve higher classification success.

Declarations

References

1. Wang, J., Zheng, H., Huang, Y., Ding, X.: Vehicle type recognition in surveillance images from labeled web-nature data using deep transfer learning. IEEE Trans. Intell. Transp. Syst. **19**(9), 2913–2922 (2018). https://doi.org/10.1109/TITS.2017.2765676

2. Al Eisaeia, M., Moridpourb, S., Tay, R.: Heavy vehicle management: restriction strategies. Transp Res Proced **21**, 18–28 (2017). https://doi.org/10.1016/j.trpro.2017.03.074

3. Anonymous.: Ultra Low Emission Zone (ULEZ). Transport For London. https://tfl.gov.uk/corporate/publications-and-reports/ultra-low-emission-zone (2021). Accessed 28 Feb 2021

4. Mishra, S., Hailu, T., Ellappan, A.V., Singh, D., Harish, R.: Avocado fruit disease detection and classification using modified SCA – PSO algorithm-based MobileNetV2 convolutional neural network. Iran J. Comput. Sci. (2022). https://doi.org/10.1007/s42044-022-00116-7

5. Wijaya, N., Mulyani, S.H., Noviadi Prabowo, A.C.: DeepDrive: effective distracted driver detection using generative adversarial networks (GAN) algorithm". Iran J. Comput. Sci. **5**(3), 221–227 (2022). https://doi.org/10.1007/s42044-022-00103-y

6. Luo, R., et al.: Dense-TNT: efficient vehicle type classification neural network using satellite imagery. IEEE Trans. Intell. Transp. Syst. (2022). https://doi.org/10.48550/arXiv.2209.13500

7. Wang, J., Cao, B., Yu, P., Sun, L., Bao, W., Zhu, X.: Deep learning towards mobile applications. Proc. Int. Conf. Distrib. Comput. Syst. **2018**, 1385–1393 (2018). https://doi.org/10.1109/ICDCS.2018.00139

8. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., et al.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)

9. 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, pp 4510–4520. arXiv.1801.04381 (2018)

10. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (2018). https://doi.org/10.1109/CVPR.2018.00907

11. Winoto, A.S., Kristianus, M., Premachandra, C.: Small and slim deep convolutional neural network for mobile device. IEEE Access **8**, 125210–125222 (2020). https://doi.org/10.1109/ACCESS.2020.3005161

12. Lee, S.H., Bang, M., Jung, K.H., Yi, K.: An efficient selection of HOG feature for SVM classification of vehicle. Proc. Int. Symp. Consum. Electron. ISCE **2015**, 14–15 (2015). https://doi.org/10.1109/ISCE.2015.7177766

13. Manzoor, M.A., Morgan, Y.: Vehicle make and model classification system using bag of SIFT features. IEEE Annu. Comput. Commun. Work. Conf. CCWC (2017). https://doi.org/10.1109/CCWC.2017.7868475

14. Seenouvong, N., Watchareeruetai, U., Nuthong, C., Khongsomboon, K., Ohnishi, N.: Vehicle detection and classification system based on virtual detection zone. Int. Jt. Conf. Comput. Sci. Softw. Eng. JCSSE (2016). https://doi.org/10.1109/JCSSE.2016.7748886

15. Zhang, B.: Reliable classification of vehicle types based on cascade classifier ensembles. IEEE Trans. Intell. Transp. Syst. **14**(1), 322–332 (2013). https://doi.org/10.1109/TITS.2012.2213814

16. Psyllos, A., Anagnostopoulos, C.N., Kayafas, E.: Vehicle model recognition from frontal view image measurements. Comput. Stand. Interfaces **33**(2), 142–151 (2011). https://doi.org/10.1016/j.csi.2010.06.005

17. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2323 (1998). https://doi.org/10.1109/5.726791

18. Rabano, S.L., Cabatuan, M.K., Sybingco, E., Dadios, E.P., Calilung, E.J.: Common garbage classification using MobileNet. IEEE. Conf. Humanoid, Nanotechnol. Inf. Technol. Commun. Control. Environ. Manag. HNICEM **2018**, 18–21 (2018). https://doi.org/10.1109/HNICEM.2018.8666300

19. Bi, C., Wang, J., Duan, Y., Fu, B., Kang, J.R., Shi, Y.: MobileNet based apple leaf diseases identification. Mob. Netw. Appl. (2020). https://doi.org/10.1007/s11036-020-01640-1

20. Ahsan, M.M., Gupta, K.D., Islam, M.M., Sen, S., Rahman, M.L., Hossain, M.S.: Study of different deep learning approach with explainable AI for screening patients with COVID-19 symptoms: using CT scan and chest X-ray image dataset. arXiv (2020). https://doi.org/10.3390/make2040027

21. Boudrioua, M.S.: COVID-19 detection from chest X-ray images using CNNs models: further evidence from deep transfer learning. SSRN Electron. J. (2020). https://doi.org/10.2139/ssrn.3630150

22. Wijaya, N., Mulyani, S.H., Anggraini, Y.W.: DeepFruits: efficient citrus type classification using the CNN. Iran J. Comput. Sci. (2022). https://doi.org/10.1007/s42044-022-00117-6

23. Swastika, W., Ariyanto, M.F., Setiawan, H., Irawan, P.L.T.: Appropriate CNN architecture and optimizer for vehicle type classification system on the toll road. J. Phys. **1196**(1), 012044 (2019). https://doi.org/10.1088/1742-6596/1196/1/012044

24. Toğaçar, M., Ergen, B., Cömert, Z.: Classification of flower species by using features extracted from the intersection of feature selection methods in convolutional neural network models. Meas. J. Int. Meas. Confed. (2020). https://doi.org/10.1016/j.measurement.2020.107703

25. Prasad, S.: Vehicle Dataset. Kaggle Platform. https://www.kaggle.com/datasets/iamsandeepprasad/vehicle-data-set (2022). Accessed 19 Sept 2022

26. Toğaçar, M., Ergen, B., Cömert, Z.: Application of breast cancer diagnosis based on a combination of convolutional neural networks, ridge regression and linear discriminant analysis using invasive breast cancer images processed with autoencoders. Med. Hypotheses **135**(3), 2020 (2020). https://doi.org/10.1016/j.mehy.2019.109503

27. Boser, B.E., Guyon, I.M., Vapnik, V.N.: Training algorithm for optimal margin classifiers. Proc Fifth Annu. ACM Work. Comput. Learn. Theory **15**, 144–152 (1992). https://doi.org/10.1145/130385.130401

28. Anthony, G., Gregg, H., Tshilidzi, M.: Image classification using SVMs: one-against-one vs one-against-all. Asian Conf. Remote Sens. **2**, 801–806 (2007)

29. Dhakshina Kumar, S., Esakkirajan, S., Bama, S., Keerthiveena, B.: A microcontroller based machine vision approach for tomato grading and sorting using SVM classifier". Microprocess. Microsyst. **76**, 3090 (2020). https://doi.org/10.1016/j.micpro.2020.103090

30. Bouchene, M.M., Boukharouba, A.: Features extraction and reduction techniques with optimized SVM for Persian/Arabic handwritten digits recognition. Iran J. Comput. Sci. **5**(3), 247–265 (2022). https://doi.org/10.1007/s42044-022-00106-9

31. Bolón-Canedo, V., Remeseiro, B.: Feature selection in image analysis: a survey. Artif. Intell. Rev. **53**(4), 2905–2931 (2020). https://doi.org/10.1007/s10462-019-09750-3

32. Liu, H., Zhou, M., Liu, Q.: An embedded feature selection method for imbalanced data classification. IEEE/CAA J. Autom. Sin. **6**(3), 703–715 (2019). https://doi.org/10.1109/JAS.2019.1911447

33. Mangal, A., Holm, E.A.: A comparative study of feature selection methods for stress hotspot classification in materials. Integr Mater Manuf Innov **7**, 87–95 (2018). https://doi.org/10.1007/s40192-018-0109-8

34. Anonymous.: Feature Selection. Scikit Learn. https://scikit-learn.org/stable/modules/feature_selection.html (2021). Accessed 26 Jan 2021

35. D. P. Kingma, J. L. Ba, Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015 Conference Track Proceedings, pp. 1–15 (2015)

36. Doğan, G., Ergen, B.: A new mobile convolutional neural network-based approach for pixel-wise road surface crack detection. Measurement **195**, 111119 (2022). https://doi.org/10.1016/j.measurement.2022.111119

37. Guo, L., Li, R., Jiang, B.: An ensemble broad learning scheme for semisupervised vehicle type classification. IEEE Trans. Neural Networks Learn. Syst. **32**(12), 5287–5297 (2021). https://doi.org/10.1109/TNNLS.2021.3083508

38. Luo, Z., et al.: MIO-TCD: a new benchmark dataset for vehicle classification and localization. IEEE Trans. Image Process. **27**(10), 5129–5141 (2018). https://doi.org/10.1109/TIP.2018.2848705

39. Chen, C.L.P., Liu, Z.: Broad learning system: an effective and efficient incremental learning system without the need for deep architecture. IEEE Trans. Neural Netw. Learn. Syst. **29**(1), 10–24 (2018). https://doi.org/10.1109/TNNLS.2017.2716952