



# Car make and model recognition system using rear-lamp features and convolutional neural networks

Michał Bularz<sup>1</sup> · Karol Przystalski<sup>1</sup>  · Maciej Ogorzałek<sup>1</sup>

Received: 29 December 2021 / Revised: 2 September 2022 / Accepted: 2 March 2023

© The Author(s) 2023

## Abstract

Recognizing cars based on their features is a difficult task. We propose a solution that uses a convolutional neural network (CNN) and image binarization method for car make and model classification. Unlike many previous works in this area, we use a feature extraction method combined with a binarization method. In the first stage of the pre-processing part we normalize and change the size of an image. The image is then used to recognize where the rear-lamps are placed on the image. We extract the region and use the image binarization method. The binarized image is used as input to the CNN network that finds the features of a specific car model. We have tested the combinations of three different neural network architectures and eight binarization methods. The convolutional neural network with parameters of the highest quality metrics value is used to find the characteristics of the rear lamps on the binary image. The convolutional network is tested with four different gradient algorithms. We have tested the method on two data sets which differ in the way the images were taken. Each data set consists of three subsets of the same car, but is scaled to different image dimensions. Compared to related works that are based on CNN, we use rear view images in different position and light exposure. The proposed method gives better results compared to most available methods. It is also less complex, and faster to train compared to other methods. The proposed approach achieves an average accuracy of 93,9% on the first data set and 84,5% on the second set.

**Keywords** Car make and model recognition · Computer vision · Convolutional neural networks · Image binarization

---

Michał Bularz, Karol Przystalski and Maciej Ogorzałek are contributed equally to this work.

✉ Michał Bularz  
mbularz95@interia.pl

Karol Przystalski  
karol.przystalski@uj.edu.pl

Maciej Ogorzałek  
maciej.ogorzalek@uj.edu.pl

<sup>1</sup> Department of Information Technologies, Jagiellonian University in Cracow, Łojasiewicza 11, Kraków, 30-348, Poland

## 1 Introduction

Vehicle make and model recognition systems (VMMR) or automated vehicle classification (AVC) support humans in extracting relevant information in many real-world applications. Compared to other pattern recognition methods, car recognition methods seem to be easy to implement. Still, there are many challenges that such systems face like the number of different vehicle types, car models and types of each model, different angle or light of the image. Until today, many methods have been proposed for car make and model recognition. The methods vary depending on the type of image used, the lighting conditions, or the angle of exposure.

Following the FBI report [1], more than 700,000 cars are stolen every year. Today, thanks to the wide-spread use of monitoring cameras and the fast growing number of applications based on machine learning, we can monitor and recognize cars. Such systems are already used by the intelligent transportation, logistics, or traffic systems (ITS). Such systems mostly use car number plate recognition. Automated license plate recognition systems (ALPR) might fail due to plate fraud or because the plates are not readable. Even if ALPR does not fail, we should use it together with CMMR systems to confirm the car details for security reasons and fraud prevention.

The goal of this paper is to build a neural network-based model where the input data is a simple binary image. We believe that additional effort in the preprocessing part should have an impact on network performance and reduce the number of parameters to train. To prove this, we divided the paper into five sections. In the next section, other papers related to the current one are described. It is followed by details of the proposed method. The six steps of the proposed model are described in detail in three subsections. The results have been explained and discussed in Section 4. followed by the conclusions in Section 5. Further ideas for extensions and possible modifications of the proposed method have been described in the last section.

## 2 Related works

There are various ways to recognize the car models in the images. Based on [2], vehicle classification systems can be divided into three groups: vehicle type, vehicle make, and vehicle make and model recognition. The most difficult group is the third group due to the complexity of the task. The known methods are able to detect the car type, model, and in many cases the exact version of the model. The researchers recognize the car model using the rear [3] or the front view of the car [4]. If possible, the car should be identified by the car plates, but in many cases it might be hard to recognize the plates, and we can recognize the car model only. Most articles focus on specific features of the car to classify the car model. Our method uses the rear view images and focuses on the rear stop lights.

Car feature-based methods have been used in several studies and are based on geographical features [5], edge-based features [4, 6], histogram of gradient (HoG) features [7, 8], contour point features [9], curvelet transform features [10], contourlet transform features [11, 12] or are combined [13–17].

Most of the works are based on shallow machine learning methods such as kNN [6, 11], SVM [16–18], or combined [14, 17, 19]. Recently, more papers have also used well-known deep neural network architectures such as ResNet [20] or CNN [21, 22].

The lighting condition considered is also an important factor in car model recognition. Most of the articles focus on daylight images. Compared to related works, most researchers

focus on the frontal view taken in daylight [4, 6, 8, 11, 14–17, 23–26]. Research is based on images of cars taken at night [3, 27–29]. Intelligent transportation systems should work throughout the day and not only during daylight. In [29] the author claims that it is more difficult to analyze night images for CMMR. In our opinion, it depends on the features that are used for analysis. The rear light when turned on is easier to extract compared to daylight images, and in this case we disagree on the complexity of night image analysis. Also, in our opinion, it is more difficult to extract and predict the car model based only on the rear light in daylight.

## 2.1 Contribution

The main contributions of this paper are the following:

- analysis and comparison of binarization methods used for car recognition,
- comparison of different neural network architectures for car model classification,
- new method that reaches a better quality compared to related solutions.

## 3 Proposed method

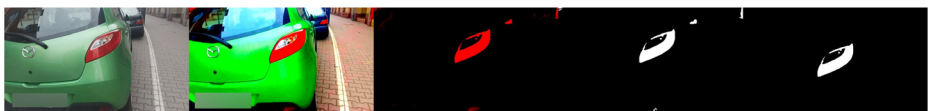
The aim of this research is to recognize the car model based on the rear lights using daylight images. Instead of a sophisticated network, we focus on the preparation of input data. The pre-processing part is the main part of the whole process, as it consists of five procedures out of six. The proposed recognition process is divided into the following steps:

1. increasing the contrast and color saturation,
2. red colour objects detection,
3. image binarization,
4. object extraction,
5. crop the image to 64x64 pixels,
6. car model classification.

In the following subsections, we explain each part of the process in more detail. The first two parts are described in the image preprocessing subsection. Steps 3–5 are described in the image binarization subsection. The last section covers the network architectures that are used in the proposed method. An overview of steps 1–5 is given in Fig. 1.

### 3.1 Image preprocessing

With a higher contrast and color saturation, the red color in the image is more emphasized and the rear lamps of the car are expressed from the rest of the photo. Any glare or other imperfections in the rear lamps are reduced and a relatively clean and strong red spot is



**Fig. 1** The process steps from the left: original image, increased contrast and color saturation image, red color object extracted image, binarized image, rear lamp cropped image

obtained in the shape of the rear lamp. The contrast and saturation factors are set empirically, respectively, as 2.0 and 4.0. Only red objects are detected in an image with increased contrast and color saturation. A mask is set on the color channel of the RGB model. As a result of this operation, the red areas retain their color and the rest of the image remains black.

### 3.2 Image Binarization

The binarization method is used in the next step. It is used to filter out the resulting noise, enhance the shape of the rear lamps, and prepare for further processing. Before binarization the image is converted to grayscale. As a result, a binarized image is obtained with the shape of the rear lamps marked and any other objects in the background that are red.

We search for the object with the largest contour (bounding box) in the binarized image. In the proposed method, we assume that the largest red object in the image is the rear lamp. Choosing the largest contour also helps if the photo shows two rear lamps, left and right. Then the one with the greatest contour is selected. The selected object with the largest outline is cropped to that outline. As a result, a sample with the desired shape of the tail lamp is obtained. The output image is a 64x64 pixel square image with a centered shape of the rear lamp.

Image binarization makes the representation of an image simpler and may highlight its features. Depending on the parameters of the binarization method, it can be a useful part of the image pre-processing and have a good impact on the classification quality.

We have checked the seven most popular binarization methods and compared the results. The binarization methods use local or global thresholding.

The Otsu method [30] belongs to the group of global thresholding methods that use image pixel clustering. This is done by dividing the grayscale image pixels represented in the histogram into two clusters, one representing the object's pixels and one describing the background pixels.

The Yen method [31] belongs to the group of algorithms that use the histogram entropy of the distribution of gray levels in the image. The algorithms of this group seek a threshold value of  $t$  that maximizes the entropy value in the binarized image, so that it transfers the most information from its prototype.

Thresholding in the Li method [32] is based on minimizing the value of the cross-entropy between the original image and the binarized image. Cross-entropy is understood here as a measure of consistency between the binarized image and the prototype.

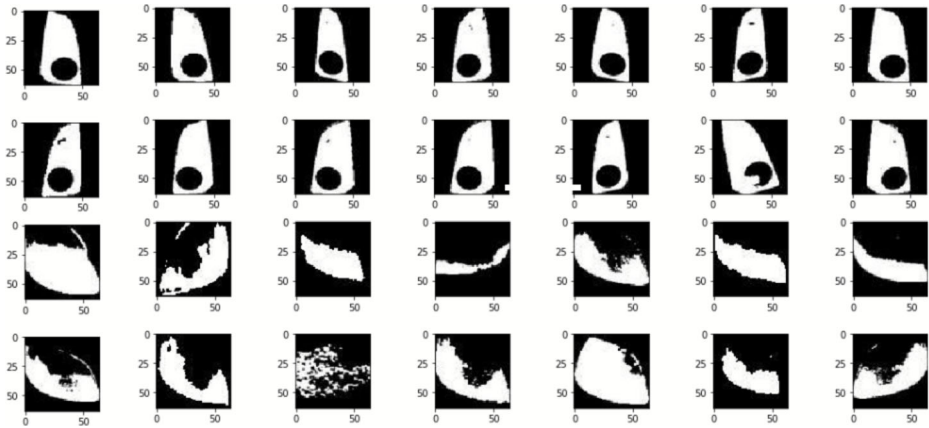
The ISODATA method [33] is also known as the Ridler-Calvard algorithm. In this method, the image is divided into two classes that correspond to:  $k_1$  object and  $k_2$  background.

The mean threshold method is one of the simplest. It works by taking the threshold value as the average of all pixel values in the grayscale image. Due to its global reach and the use of an average, it can often be inaccurate.

The Niblack method [34] is an example of a local thresholding method. The threshold is set independently for each pixel based on knowledge of the mean and standard deviation of the pixel values surrounding the pixel currently considered with coordinates  $(i, j)$ .

The Sauvola method [35] is a modification of the local Niblack thresholding method, but the method is known to be faster and in some cases gives better results than its predecessor.

All thresholds were empirically selected for each image and method. The results are given in Fig. 2.



**Fig. 2** Examples of image binarization using different methods of Opel Agila A (first two rows) and Peugeot 206 (last two rows). In columns binarization methods used: Otsu, mean, Yen, ISOData, Li, Sauvols, Niblack

### 3.3 Network architecture

To check how the selected neural network architecture affects its generalization ability, we decided to investigate three architectures:

- Convolutional network with 3 convolution blocks.
- Convolutional network with 2 blocks, a network with a reduced number of convolutional blocks. All its other elements, such as the number of layers in a convolutional block, the number and size of filters, the use of connecting layers and *dropout*, remained unchanged.
- Ordinary dense neural network, an ordinary, non-volutionary neural network with two hidden dense layers of 1024 neurons each. Each such layer is followed by *dropout* with a probability of 0.5.

A convolutional neural network consisting of 3 convolutional blocks was used. There are two convolutional layers in each of them, the first of which uses zero-padding. At the end of each such block, there are layers that connect max-pooling to a 2x2 window size, followed by dropout with a probability of 0.5, preventing overfitting. Each convolution layer uses a 3x3 filter size. The number of filters changes with successive blocks - in the first one it is 32 filters, in the second 64 and in the last 128. After the convolutional blocks, there is a classifier consisting of two dense hidden layers, consisting of 1024 neurons each. Each hidden layer is followed by the dropout with a probability of 0.5. The max-pooling connecting layers are marked with green triangles between the blocks.

The selection of network parameters, such as the gradient algorithm or the value of the  $\eta$  learning coefficient, can have a noticeable impact on the results achieved. To check this, we decided to test three additional gradient algorithms with different values of  $\eta$ :

- Stochastic Gradient Descent - with parameter  $\eta = 0.001$  and momentum = 0.8,
- Adam - with parameter  $\eta = 0.0001$ ,
- AdaGrad - with parameter  $\eta = 0.001$ .

## 4 Results

The results are divided according to the image segmentation methods, the gradient descent algorithm used, and the network architecture. The first part deals with the data sets that are used in the proposed method. The set of results presented in the following sections gives a good overview of the combination of methods used to find the network with the highest quality metrics values. The quality metrics that are used are: accuracy, specificity, precision, and  $F_1$  score. The accuracy is a well-known quality metric defined as:

$$ACC = \frac{TP + TN}{TP + FN + FP + TN}. \quad (1)$$

The specificity is a metric that explains how good a given method is at finding positive cases. In proposed methods, it explains how good the method is in finding car models. It is also known as PPV and is defined as:

$$PPV = \frac{TP}{TP + FP}. \quad (2)$$

$$TNR = \frac{TN}{TN + FP}. \quad (3)$$

$$F_1 - score = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR}. \quad (4)$$

### 4.1 Data set

Due to the specific properties of the problem and the selected classification criterion understood as the shape of the rear lamp of the vehicle, it is required that the images in the training data set show only the rear of the vehicle at different angles. The same condition also applies to the test set. For this reason, we decided to create our own data sets.

Data were collected over a period of several months as follows. With a smartphone camera, the rear of surrounding cars was recorded at different angles during the day for about 30 seconds. During recording, we ensured that at least one rear lamp was clearly visible. The recordings were made only during daylight due to the fact that the vehicle's off-lights were clearly visible then. Then, all the frames of the video were removed from the recording, obtaining several hundreds images of the back of the car. Each model was recorded at different times of the day, under different lighting conditions, and in different places. Most of the models were recorded many times to increase the diversity of the data set. We obtained more than 300 thousand images in total.

The target set of data on which the neural network has been trained and tested is a set of images after initial processing, taking into account the selected binarization method and after extraction of the main feature. Each sample is a binarized image of the default size of 64x64 pixels with the tail lamp shape marked. Due to the fact that it is a binarized image, it can be interpreted as a 64x64 matrix of the extracted lamp shape.

The ability of the network to generalize has been assessed through double cross-validation on the basis of two data sets: CarBinLamps and mixed. The CarBinLamps data set is a set of data collected with a smartphone. There are 100 randomly selected photos from the whole pool for each car model. We have 100 different car make and model classes. This makes a total of 10 thousand images in the CarBinLamps data set. We have shared the dataset as a Kaggle dataset and it is available at <https://www.kaggle.com/michabularz/>

[car-tailights-shapes](#). The double cross-validation approach solves the problem of an unbalanced data set. We used the ratio of 80% of the training data set to 20% testing. The CarBinLamps collection does not include all possible cases and angles at which a photograph of the rear of a car can be taken. As mentioned above, the photos from this data set satisfy a number of assumptions. This data set will be used to check the ability of the network to generalize and get the correct predictions when the user is instructed in advance on how to take a picture of the car to get the correct result.

The CarBinLamps data set does not cover all cases, therefore, we decided to refill it. For this purpose, a set of selected images from the Google Image search engine network was used, showing models of cars with visible rear lamps at different angles, sometimes much larger than in the CarBinLamps set. Each car model has approximately 10 images of various sizes from the above-mentioned search engine. These additional images do not contain red cars. This set was processed in the same way as the CarBinLamps set. Ultimately, the mixed collection consists of 4,000 images. For each car model, there are 20 samples from the CarBinLamps set and 20 samples based on photos from the Google Graphics search engine. This data set is used to check how the network behaves when the user is not precisely instructed how to take a photo of the back of the car to get the correct result. We used the ratio of 80% of the training data set to 20% testing.

## 4.2 Binarization methods

Each binarization method has its own dedicated set of data processed with its use. Each sample obtained in the preprocessing process is copied and flipped horizontally (mirror image) to increase the diversity of the data.

In order for the effectiveness tests of the binarization methods used to be reliable, we decided to choose one neural network architecture and the same parameters (selection of the gradient algorithm and values of the learning factor) for all the tests regarding the binarization methods given in Table 1.

**Table 1** Average quality metrics values set by binarization methods

Binarization method	Dataset	Accuracy	Sensitivity	Specificity	<i>F1-score</i>
<b>Otsu</b>	Mixed	<b>0.762</b>	0.763	0.804	0.762
	CarBinLamps	<b>0.851</b>	0.851	0.877	0.851
<b>Mean</b>	Mixed	<b>0.622</b>	0.624	0.693	0.625
	CarBinLamps	<b>0.742</b>	0.742	0.793	0.745
<b>Yen</b>	Mixed	<b>0.597</b>	0.596	0.67	0.604
	CarBinLamps	<b>0.711</b>	0.71	0.762	0.71
<b>ISODATA</b>	Mixed	<b>0.75</b>	0.751	0.799	0.752
	CarBinLamps	<b>0.852</b>	0.852	0.879	0.852
<b>Li</b>	Mixed	<b>0.69</b>	0.692	0.742	0.692
	CarBinLamps	<b>0.806</b>	0.806	0.84	0.805
<b>Sauvola</b>	Mixed	<b>0.661</b>	0.662	0.72	0.66
	CarBinLamps	<b>0.766</b>	0.766	0.819	0.774
<b>Niblack</b>	Mixed	<b>0.52</b>	0.523	0.624	0.538
	CarBinLamps	<b>0.627</b>	0.627	0.738	0.649

Categorical cross-entropy is used as the error function due to the multiclass problem. RMSProp with the learning coefficient  $\eta = 0.0001$  is empirically selected as the gradient algorithm.

Looking at the obtained results, it can be seen that depending on what method of binarization is used during the initial data processing, the quality of the classification may differ significantly. The highest classification accuracy on the CarBinLamps set was obtained using the ISODATA binarization method (0.852), while on the mixed set - using the Otsu binarization method (0.762). It is worth noting that both of these methods achieved very similar values of the metrics on the corresponding data sets. The similarity in the obtained results can be explained by looking at the similarity in the representation of the data obtained by these binarization methods. The images created after binarization of Otsu and ISODATA are very similar to each other.

The lowest classification accuracy was achieved on both sets in the case of the Niblack binarization method (0.52 in the mixed and 0.627 in the CarBinLamps). Such a poor result can be explained mainly by the representation of the data obtained by this binarization method, because the obtained samples are full of noise and imperfections. Furthermore, it can be seen that this method tends to accentuate various noise contours. However, the desired shape of the rear lamp is not clean and often even omitted. This behavior can be explained by the fact that the Niblack binarization method (similar to that of Sauvola) is dedicated to the problems of text binarization and working with black letters on a light background, and the problem discussed in this paper does not belong to this class of problems. The Sauvola binarization method was created as an improvement of the Niblack method, and the difference in the results achieved by these methods on both data sets (the difference in accuracy is about 0.14) seems to confirm this. In the case of the Sauvola method, we obtain significantly better metric values than in the case of Niblack binarization. This is also due to the fact that the samples obtained by this method are much more noise-free and clean.

### 4.3 Network architectures

All tests were carried out on two data sets using Otsu binarization. Five-fold cross-validation was used to determine the generalizability of the network. The average results obtained in this way are presented in the Table 2.

Based on the obtained results, it is clear that the selection of the appropriate network architecture is of great importance for the generalization ability of the network. An accuracy of 0.727 on a mixed dataset is a very good result for the DNN architecture, although the result achieved by the convolutional network is noticeably better. The advantage of this

**Table 2** Average quality metrics values for different network architectures

Network architecture	Dataset	Sensitivity	Specificity	Precision	<i>F1-score</i>
<b>CNN with 3 levels</b>	Mixed	<b>0.762</b>	0.763	0.804	0.762
	CarBinLamps	<b>0.851</b>	0.851	0.877	0.851
<b>CNN with 2 levels</b>	Mixed	<b>0.814</b>	0.814	0.84	0.814
	CarBinLamps	<b>0.909</b>	0.909	0.92	0.909
<b>DNN</b>	Mixed	<b>0.727</b>	0.73	0.761	0.73
	CarBinLamps	<b>0.841</b>	0.841	0.858	0.841



solution is a faster training process due to the smaller number of weights and the less complicated learning process. The use of a convolutional network with a reduced number of convolutional blocks gives noticeably better results, where the value of 0.814 is achieved on the mixed set. Deeper convolution layers extract high-level features from the data. However, the use of too many convolutional layers causes the mentioned high-level features to be distorted, which worsens the ability of the network to generalize. The images in the data set are simplified to the form of binary matrices. Here we should look for the reason why the reduced convolutional network fared noticeably better than the network with an additional convolutional block.

The best accuracy on the mixed set is achieved with the architecture of a convolution neural network with 2 convolution blocks (0.814); therefore, tests in subsequent subsections are conducted with it.

#### 4.4 Gradient descent algorithm

Each of the above mentioned networks has the same parameters in the form of the RMSProp gradient algorithm with the learning coefficient  $\eta = 0.0001$  and the error function in the form of categorical cross entropy. Thanks to this approach, the results obtained are more objective.

All of the above tests were carried out using 5-fold cross-validation on two data sets. A convolution network architecture with two blocks was selected due to the fact that the best accuracy is achieved by this network architecture. The results are given in the Table 3.

The obtained results show that selection of the appropriate gradient algorithm and learning coefficient have a large impact on the ability of the network to generalize. The best result was achieved for the Adam algorithm, where the accuracy value is 0.845 on the mixed set. The AdaGrad algorithm is the worst, reaching a 0.748 accuracy on a mixed set.

The choice of the learning coefficient is also important because if the value is too low, the learning process will be slow and if the value is too high, the learning process may be completely disrupted due to too aggressive changes in the weight of neurons during learning. The optimal values of this coefficient for each algorithm tested have been chosen empirically.

Due to the fact that the best accuracy was achieved with the Adam gradient algorithm, the tests in the next subsection are performed with it.

**Table 3** Average quality metrics values for different gradient descent algorithms

Gradient algorithm	Data set	Accuracy	Sensitivity	Precision	<i>F1-score</i>
<b>RMSProp</b> ( $\eta = 0.0001$ )	Mixed	<b>0.814</b>	0.814	0.84	0.814
	CarBinLamps	<b>0.909</b>	0.909	0.92	0.909
<b>SGD</b> ( $\eta = 0.001, \gamma = 0.8$ )	Mixed	<b>0.795</b>	0.796	0.82	0.794
	CarBinLamps	<b>0.902</b>	0.902	0.91	0.9
<b>Adam</b> ( $\eta = 0.0001$ )	Mixed	<b>0.845</b>	0.845	0.861	0.842
	CarBinLamps	<b>0.939</b>	0.943	0.939	0.939
<b>AdaGrad</b> ( $\eta = 0.001$ )	Mixed	<b>0.748</b>	0.749	0.775	0.745
	CarBinLamps	<b>0.873</b>	0.873	0.884	0.87

**Table 4** Average quality metrics values for different image sizes

	64x64	32x32	16x16
<b>Accuracy</b>	0.845	0.798	0.594
	<b>0.939</b>	0.906	0.745
<b>Sensitivity</b>	0.845	0.799	0.597
	0.943	0.906	0.745
<b>Specificity</b>	0.861	0.823	0.623
	0.939	0.913	0.765
<b>F<sub>1</sub>-score</b>	0.842	0.798	0.597
	0.939	0.904	0.732

## 4.5 Image size

By default the dimensions of images in the dataset are 64 by 64 pixels, but it is worth checking if changing their size will affect the network's ability to generalize. For this purpose, tests were carried out with 5-fold cross-validation on two data sets for the image dimensions 32x32 and 16x16 pixels. All tests were performed for the Otsu binarization method with the convolutional network architecture composed of two blocks and with the Adam gradient algorithm ( $\eta = 0.0001$ ). The test results are shown in the Table 4.

It is clear here that as the sample size decreases, the accuracy also drops as the other metrics do. The obtained results show that the smaller the sample size in the data set, the less information it contains. Thus, with an insufficiently large sample of data, obtaining optimal classification results may be difficult as the network then receives less information that can be used in the training process. On the other hand, the undoubted advantages of reducing the sample size are a noticeable increase in efficiency and shortening the network training time.

## 5 Conclusions

The conclusions can be divided into two parts. In the first part, we compare the proposed method with related work and show the advantages of it. Our method also has a few major limitations, which are described in the next part of this section.

### 5.1 Comparison

The data set used in this paper is the largest taillight set used for vehicle make and model recognition so far. Based on [2] the CompCars data set consists of 136 thousand images of 1716 car models. The CompCars data set has only 3563 taillight images, but it also has 44663 rear and side-rear images of the car. The BoxCars data set consists of 63750 car images, each with a rear-view image. This data set has 126 car models. The Stanford-Cars data set consists of 16185 images and VeRi-776 of 49357 images of 776 vehicles. BoxCars, CompCar, and VeRi-776 data sets can be used for further analysis of the method presented in this paper. Our data sets consist of 300k images in total of 100 different car models. The VERI-Wild data set 12 million vehicles taken from CCTV. Compared to previous data sets, on one image multiple cars are visible.

**Table 5** Comparison of proposed method with recently related methods

	Accuracy	View	Dataset and car models	Method
Manzoor et al. [36]	97.89%	Front	About 6000 and 35 models	SVM
Kim et al. [37]	78%	Front	300 images	SRGAN
Lee et. al [38]	94.23%	Front	Stanford-Cars	Residual SqueezeNet
Ghassemi et al. [39]	97.8 %	All angles	CompCars	WideResNet
Xiang et al. [40]	97.07%–99.29%	Front	CompCars	ResNet101-GRPSP
Khorramshahi et al. [41]	88.97%	Front and rear	VeRi-776	Attention Network
Corrales et al. [42]	97.62%	All angles	CompCars	CNN with 2 fine tuning steps
Wang et al. [43]	90.51%	Front	163	DCNN
Lu and Huang [44]	92.53%	Front	12,238	multiclass SVM
Jamil et al. [45]	98.22%	Front	About 6000 images and 29 models	SVM
Fomin et al. [46]	92.6%	All angles	16185 images and 169 classes	VGG-16, Resnet, Yolo
Llorca et al. [18]	93.75%	Rear	1.342 images	Linear SVM
Boonsim and Prakoonwit [29]	93.8%	Rear	766 images	Support vector machine, decision tree, and k-nearest neighbors
Abbas et al. [47]	97.3%	Front	About 3000 images and about 20 models	kNN
Dehkordi [48]	89.5%	All angles	7000 images	BoW with SVM
Sochor et al. [22]	85%	All angles	BoxCars and CompCars	CNN
Ni et al. [49]	96.2%	All angles	VERI-Wild	ResNet50
Park et al. [50]	84.6–98.0%	Front	Custom	Yolo v4
Proposed method	93.9%	Rear	CarBinLamps	CNN

The comparison of methods can be made with quality metrics and the data set that was used for training and testing. We measured the quality of our method with four metrics: accuracy, sensitivity, specificity, and the  $f_1$  score. The authors of the methods we compared have not always used the same quality metrics. To simplify the comparison, we used the accuracy. Our method should be compared with other methods in which rear-view images are used, but the minority of published papers use this kind of image. In Table 5 the most recent articles are shown. The BoxCars data set together with the CompCars data set are used in [22]. In this case, not only rear view images were used. The accuracy varies depending on the data sets used and the size of the set. The lowest accuracy reached is 0.761% and the highest is 0.85%. A comparison is also made for different class sample sizes and the results vary from 0.731% to 0.832%. A higher accuracy was achieved in [18], but the

data set is limited to 1342 images and only 8 different car models. The accuracy reached a level of 93.75%. In [29] the classification is performed under limited light conditions. The data set consists of 766 images and 421 car models. In our opinion, it is easier to achieve higher accuracy using images with limited lighting conditions at night because features such as lamps are easier to extract. The accuracy achieved at the level of 93.8% is close to our results. The best results were achieved by [40, 50] with an accuracy of almost 100% in one of the cases. Both papers analyze the front view images only. The best results where the rear view is taken were achieved by [39]. Many papers with high accuracy use a deep neural network. In our methods, instead of a complex neural network, a few image pre-processing methods are used to simplify the input image.

We proved that comparing the complexity of the method, our method is simpler than most of the methods given in the Table 5. Our network consists of just a few layers, and the input image is simplified to a binary image. Even in such a simple network, our model outperforms many other solutions presented in Table 5. Seeing the current trend of increasing the number of parameters to be trained in network-based models, our approach shows that this trend might not be the best solution in some cases. Work on pre-processing can increase accuracy while decreasing the complexity of the model. According to [51], the complexity of our methods and the effort to train such a model are lower and, in our opinion, a more efficient approach.

## 5.2 Limitations

The method chosen to extract the shape of the rear lamp as the main characteristic of the car model is not perfect and, in some cases, may fail. First of all, the method fails for red cars. The color of the car blends with the color of the tail lights. Processing a red car image produces the entire contour of the vehicle, not the shape of a tail lamp. As mentioned above, such cases have been omitted from this work.

The accuracy of the resulting shape of the rear lamp depends on many factors, including lighting conditions - mainly reflections, which sometimes disturb the regularity of the shape - and on the binarization method used. Depending on the binarization method chosen, different representations of the same lamp are obtained. Figure 2 shows a comparison of the binarization methods tested in the example of a selected vehicle model.

In some cases, parts of the rear lamp are obtained. This problem applies mainly to situations where the rear lights are oblong or split into pieces with a thin section connecting them together. As a result of red detection and the binarization method used, thin fragments are sometimes lost, causing the rear lamp to split into two separate objects and the larger one is selected.

## 6 Further work

In the first place, we should focus on the limitations and try to reduce the impact of each on the final result. Lamps can be found by other methods than color. This can also solve the problem with the second obstacle, which is sometimes incorrect recognition of the rear lamp with the current method. In the second step, the method can be extended with other parts analysis. Possibly analyzing the exhaust pipe, air intake, or headlights might increase accuracy and other metrics. A further research can take into account more complex architectures such as Inception ResNet or MobileNet. Comparing a simple CNN network proposed

in this paper with a more complex one can result in an additional conclusion on the importance of network complexity. Networks such as Yolo v5 after modification are made for binary image input should be considered.

**Data Availability** The datasets generated during and/or analysed during the current study are available in the CarBinLamps repository, <https://www.kaggle.com/michabularz/car-taillights-shapes>.

## Declarations

**Competing interests** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. FBI.gov Motor Vehicle Theft
2. Boukerche A, Siddiqui AJ, Mammeri A (2017) Automated vehicle detection and classification: models, methods, and techniques. *ACM Comput Surv* 50:1–39
3. Boonsim N, Prakoonwit S (2014) An algorithm for accurate taillight detection at night. *Int J Comput Appl* 100:31–35. <https://doi.org/10.5120/17579-8345>
4. Pearce G, Pears N (2011) Automatic make and model recognition from frontal images of cars. In: 2011 8th IEEE International conference on advanced video and signal based surveillance (AVSS), pp 373–378, <https://doi.org/10.1109/AVSS.2011.6027353>
5. Daya B, Akoum AH, Chauvet P (2010) Identification system of the type of vehicle. In: 2010 IEEE Fifth international conference on bio-inspired computing: theories and applications (BIC-TA), pp 1607–1612, <https://doi.org/10.1109/BICTA.2010.5645260>
6. Monroe DT, Madden MG (2005) Multi-class and single-class classification approaches to vehicle model recognition from images. In: Proceedings of the Irish conference. artificial intelligence cognitive science (AICS), pp 93–102
7. Kamal I (2012) Car recognition for multiple data sets based on histogram of oriented gradients and support vector machines. In: 2012 International conference on multimedia computing and systems, pp 328–332, <https://doi.org/10.1109/ICMCS.2012.6320284>
8. Lee S, Gwak J, Jeon M (2013) Vehicle model recognition in video. *Int J Signal Process Image Process Pattern Recog* 6:175–184
9. Negri P, Clady X, Milgram M, Poulénard R (2006) An oriented-contour point based voting algorithm for vehicle type classification. vol. 1, pp 574–577
10. Mohammad Kazemi F, Samadi S, Pourreza H, Akbarzadeh-T M-R (2007) Vehicle recognition using curvelet transform and svm. vol. 0, pp 516–521
11. Clady X, Negri P, Milgram M, Poulénard R (2008) Multi-class vehicle type recognition system. vol. 5064, pp 228–239
12. Rahati S, Moravejian R, Kazemi EM, Kazemi FM (2008) Vehicle recognition using contourlet transform and svm. In: Fifth international conference on information technology: new generations (itng 2008), pp 894–898
13. Arzani M, Jamzad M (2010) Car type recognition in highways based on wavelet and contourlet feature extraction. pp 353–356
14. Zhang B (2013) Reliable classification of vehicle types based on cascade classifier ensembles. *IEEE Trans Intell Trans Syst* 14:322–332. <https://doi.org/10.1109/TITS.2012.2213814>
15. Psyllos A, Anagnostopoulos CN, Kayafas E (2011) Vehicle model recognition from frontal view image measurements. *Comput Stand Interfaces* 33(2):142–151. <https://doi.org/10.1016/j.csi.2010.06.005>

16. Baran R, Glowacz A, Matiolanski A (2015) The efficient real- and non-real-time make and model recognition of cars. *Multimedia Tools and Applications* 74
17. Hsieh J-W, Chen L-C, Chen D-Y, Cheng S-C (2013) Vehicle make and model recognition using symmetrical surf. pp 472–477
18. Fernández-Llorca D, Colas D, Garcia Daza I, Sotelo M-A (2014) Vehicle model recognition using geometry and appearance of car emblems from rear view images. In: 2014 17th IEEE International conference on intelligent transportation systems, ITSC 2014, pp 3094–3099, <https://doi.org/10.1109/ITSC.2014.6958187>
19. He H, Shao Z, Tan J (2015) Recognition of car makes and models from a single traffic-camera image. *IEEE Trans Intell Transp Syst* 16(6):3182–3192
20. Ma X, Boukerche A (2020) An ai-based visual attention model for vehicle make and model recognition. In: 2020 IEEE Symposium on computers and communications (ISCC), pp 1–6
21. Krause J, Jin H, Yang J, Fei-Fei L (2015) Fine-grained recognition without part annotations. In: 2015 IEEE Conference on computer vision and pattern recognition (CVPR), pp 5546–5555
22. Sochor J, Herout A, Havel J (2016) Boxcars: 3d boxes as cnn input for improved fine-grained vehicle recognition. In: 2016 IEEE Conference on computer vision and pattern recognition (CVPR), pp 3006–3015
23. Zafar I, Edirisinghe EA, Acar S, Bez HE (2007) Two-dimensional statistical linear discriminant analysis for real-time robust vehicle-type recognition. *Int Soc Opt Photon* 6496:9–16. <https://doi.org/10.1117/12.704592>
24. Varjas V, Tanács A (2013) Car recognition from frontal images in mobile environment. pp 812–816
25. Saravi S, Edirisinghe EA (2013) Vehicle make and model recognition in cctv footage. In: 2013 18th International conference on digital signal processing (DSP), pp 1–6
26. Fraz M, Edirisinghe EA, Sarfraz MS (2014) Mid-level-representation based lexicon for vehicle make and model recognition. In: 2014 22nd International conference on pattern recognition, pp 393–398
27. Wang C, Huang S-S, Fu L-C, Hsiao P-Y (2005) Driver assistance system for lane detection and vehicle recognition with night vision. pp 3314–3319
28. Gormer S, Muller D, Hold S, Meuter M, Kummert A (2009) Vehicle recognition and ttc estimation at night based on spotlight pairing. pp 1–6
29. Boonsim N, Prakoonwit S (2017) Car make and model recognition under limited lighting conditions at night. *Pattern Anal Appl* 20(4):1195–1207
30. Otsu N (1979) A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybernet* 9(1):62–66
31. Yen J-C, Chang F-J, Chang S (1995) A new criterion for automatic multilevel thresholding. *IEEE Trans Image Process* 4(3):370–378
32. Li CH, Tam PKS (1998) An iterative algorithm for minimum cross entropy thresholding. *Pattern Recogn Lett* 19(8):771–776
33. Velasco FD (1980) Thresholding using the isodata clustering algorithm. *IEEE Trans Syst Man Cybernet* 10(11):771–774
34. Niblack W (1985) An introduction to digital image processing
35. Sauvola J, Pietikäinen M (2000) Adaptive document image binarization. *Pattern Recogn* 33(2):225–236
36. Manzoor MA, Morgan Y, Bais A (2019) Real-time vehicle make and model recognition system. *Mach Learn Knowl Extraction* 1:611–629. <https://doi.org/10.3390/make1020036>
37. Kim J, Lee J, Song K, Kim Y-S (2019) Vehicle model recognition using srGAN for low-resolution vehicle images. In: Proceedings of the 2nd international conference on artificial intelligence and pattern recognition. AIPR '19, Association for Computing Machinery, pp 42–45
38. Lee HJ, Ullah I, Wan W, Gao Y, Fang Z (2019) Real-time vehicle make and model recognition with the residual squeezeNet architecture. *Sensors* 19:982. <https://doi.org/10.3390/s19050982>
39. Ghassemi S, Fiandrotti A, Caimotti E, Francini G, Magli E (2019) Vehicle joint make and model recognition with multiscale attention windows. *Signal Process Image Commun* 72:69–79
40. Xiang Y, Fu Y, Huang H (2019) Global relative position space based pooling for fine-grained vehicle recognition. *Neurocomputing* 367:287–298
41. Khorramshahi P, Kumar A, Peri N, Rambhatla S, Chen J-C, Chellappa R (2019) A dual-path model with adaptive attention for vehicle re-identification. pp 6131–6140
42. Corrales Sánchez H, Fernández-Llorca D, Vigre S, Quintanar Pascual A, Lorenzo Díaz J, Hernández N (2020) CNNs for fine-grained car model classification. pp 104–112
43. Wang H, Peng J, Zhao Y, Fu X (2020) Multi-path deep cnns for fine-grained car recognition. *IEEE Trans Veh Technol* 69(10):10484–10493
44. Lu L, Huang H (2020) Component-based feature extraction and representation schemes for vehicle make and model recognition. *Neurocomputing* 372:92–99

45. Jamil AA, Hussain F, Yousaf M, Butt AM, Velastin S (2020) Vehicle make and model recognition using bag of expressions. *Sensors (Basel Switzerland)* p 20
46. Fomin I, Nenahov I, Bakhshiev A (2020) Hierarchical system for car make and model recognition on image using neural networks. In: 2020 International conference on industrial engineering, applications and manufacturing (ICIEAM), pp 1–6
47. Abbas A, Sheikh U, Mohd M (2020) Recognition of vehicle make and model in low light conditions. *Bulletin of Electrical Engineering and Informatics* p 9
48. Asgarian Dehkordi R, Khosravi H (2020) Vehicle type recognition based on dimension estimation and bag of word classification. *J AI Data Min* 8(3):427–438
49. Ni X, Huttunen H (2021) Vehicle attribute recognition by appearance: computer vision methods for vehicle type, make and model classification. *Journal of Signal Processing Systems* 93
50. Park S-H, Yu S-B, Kim J-A, Yoon H (2022) An all-in-one vehicle type and license plate recognition system using yolov4. *Sensors* 22:921. <https://doi.org/10.3390/s22030921>
51. Dodge J, Gururangan S, Card D, Schwartz R, Smith NA (2019) Show your work: improved reporting of experimental results

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.