Journal of
Electronic Imaging

JElectronicImaging.org

Semiautomatic training and evaluation of a learning-based vehicle make and model recognition system

Matthijs H. Zwemer
Guido M. Y. E. Brouwers
Rob G. J. Wijnhoven
Peter H. N. de With

SPIE. IS&T imaging.org

Matthijs H. Zwemer, Guido M. Y. E. Brouwers, Rob G. J. Wijnhoven, Peter H. N. de With, "Semiautomatic training and evaluation of a learning-based vehicle make and model recognition system," *J. Electron. Imaging* **27**(5), 051225 (2018), doi: 10.1117/1.JEI.27.5.051225.

Semiautomatic training and evaluation of a learning-based vehicle make and model recognition system

Matthijs H. Zwemer,[a,b,*] **Guido M. Y. E. Brouwers,**[b] **Rob G. J. Wijnhoven,**[b] **and Peter H. N. de With**[a]
[a]Eindhoven University of Technology, Department of Electrical Engineering, VCA Research Group (SPS-VCA), Eindhoven, The Netherlands
[b]ViNotion B.V., Eindhoven, The Netherlands

Abstract. We describe a system for vehicle make and model recognition (MMR) that automatically detects and classifies the make and model of a car from a live camera mounted above the highway. Vehicles are detected using a histogram of oriented gradient detector and then classified by a convolutional neural network (CNN) incorporating the frontal view of the car. We propose a semiautomatic data-selection approach for the vehicle detector and the classifier, by using an automatic number plate recognition engine to minimize human effort. The resulting classification has a top-1 accuracy of 97.3% for 500 vehicle models. This paper presents a more extensive in-depth evaluation. We evaluate the effect of occlusion and have found that the most informative vehicle region is the grill at the front. Recognition remains accurate when the left or right part of vehicles is occluded. The small fraction of misclassifications mainly originates from errors in the dataset, or from insufficient visual information for specific vehicle models. Comparison of state-of-the-art CNN architectures shows similar performance for the MMR problem, supporting our findings that the classification performance is dominated by the dataset quality. [DOI: 10.1117/1.JEI.27.5.051225]

Keywords: traffic; convolutional neural network classification; vehicle make and model; dataset generation; occlusion.

Paper 180038SS received Jan. 11, 2018; accepted for publication Apr. 24, 2018; published online May 10, 2018.

1 Introduction

There are thousands of surveillance cameras installed along highways that are mainly used for traffic management and law enforcement. Continuous manual inspection is not feasible, as this requires enormous manual effort involving high costs. Automatic visual interpretation enables detection, tracking, and classification of all traffic. One specifically important concept is visual make and model recognition (MMR). Make and model information of vehicles can be used to find vehicles with stolen license plates, when comparing the observed vehicle model information with the registered information associated with the license plate. An additional application is to find specific vehicles after a crime when only a vehicle description is available without the license-plate number. In such cases, make and model of the vehicle need to be obtained visually. These challenges are the focal point of this paper.

Recognition of the vehicles in the above applications is now performed by an automatic number plate recognition (ANPR) system in combination with a lookup in the national vehicle registration database. Although this works for most cases, it is easy to circumvent this database matching by altering the license plates. Moreover, it does not work for vehicles without a license plate, foreign vehicles, or for motorcycles (when considering a frontal viewpoint).

The objective of this paper is therefore to solve the mismatch and missing license plates cases with an accurate visual analysis system. To this end, we present an MMR system developed for the Dutch National Police, in which vehicles are observed from a camera mounted in an overhead sign structure on the highway, with the purpose to extract accurate make and model information. The extracted information may be combined with existing ANPR information. The system implementation has a focus on observing a single lane (see Fig. 1). This existing camera is used to feed the training process of our recognition system. The recognition model is trained to recognize vehicles from a large training set of vehicle images and make and model labels. Due to bandwidth restrictions between the camera (online) and our training and testing facilities (offline), we have to optimize the gathering of training and testing samples. Another challenge is the automated handling of new and rare vehicle models as registered in the vehicle registration database, for which it is hard to collect training and testing images. For these reasons, we propose a semiautomatic system to create a vehicle dataset. The sampling and their annotation in this system are automated, while the updated training still needs manual control. This approach enables the construction of an initial dataset and allows to incrementally collect new vehicle samples over time, so that the best system performance is ensured at all moments.

The MMR system consists of a detection and a classification stage, to localize and recognize vehicles in a full-frontal view. The aim is to find the vehicle make and model information without being dependent on an ANPR system. Our two-stage approach enables detection of vehicles in every video frame and performs classification once a vehicle is found. This paper extends our initial work[1] by providing extensive insight in our MMR classification performance and discussing the evaluation of the MMR system in high detail. First, a comparison between different convolutional neural

*Address all correspondence to: Matthijs H. Zwemer, E-mail: m.zwemer@tue.nl

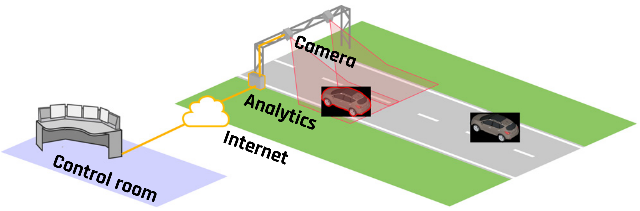

Fig. 1 The roadside analysis system and traffic control room for our MMR system.

networks for vehicle model classification is reported. Second, we give more insight into the classification performance by finding the most informative region for MMR classification and measure the robustness against occlusions. Third, the false classifications are further investigated to find shortcomings in the system and information handling.

The structure of the paper is as follows. We commence with an overview of related work in Sec. 2. Then the two-stage detection and classification system is described in Sec. 3. The semiautomatic gathering of the dataset is explained in Sec. 4 and a detailed evaluation of our system on the dataset is discussed in Sec. 5. After our in-depth evaluation of the vehicle MMR system, we discuss the application for law enforcement to assist the police with the problem of vehicle theft and evaluate computation times of the real-time system in Sec. 6.

2 Related Work

Our vehicle recognition system consists of a detection and a classification stage, to localize and recognize vehicles in a full-frontal view. The first detection stage can be solved with different approaches. The full vehicle extent is detected using frame differencing by Ren and Lan[2] or background subtraction by Prokaj and Medioni.[3] Siddiqui et al.[4] and Petrović and Cootes[5] extended detections from a license-plate detector. Wijnhoven and de With[6] proposed a histogram of oriented gradient (HOG)[7] to obtain contrast-invariant detection. Recent work by Zhou et al.[8] reports on a convolutional neural network (CNN) to obtain accurate vehicle detection. When the vehicle is detected, the vehicle region of the image is used as input for the classification task of MMR.

Image classification has been also broadly reported. CNNs are state-of-the-art for image classification and originate by work from LeCun et al.[9] and gained popularity by Krizhevsky et al.,[10] who used a CNN (AlexNet) to achieve top performance in the 1000-class ImageNet Challenge.[11] For MMR, Ren and Lan[2] proposed a modified version of AlexNet to achieve 98.7% using 233 vehicle models in 42,624 images. Yang et al.[12] published the CompCar dataset which contains different car views, different internal, and external parts, and 45,000 frontal images of 281 different models. They showed that AlexNet[10] obtains comparable performance to the more recent Overfeat[13] and GoogLeNet[14] CNN models (98.0% versus 98.3% and 98.4%, respectively). Siddiqui et al.[4] showed that for small-scale classification problems, Bag of SURF features achieve an accuracy of 94.8% on a vehicle dataset containing 29 classes in 6639 images.

Other work extends full-frontal recognition toward more unconstrained viewpoints. Sochor et al.[15] used a three-dimensional (3-D) box model to exploit viewpoint variation, Prokaj and Medioni[3] employed structure from motion to align 3-D vehicle models with images, and Dehghan et al.[16] achieved good recognition results but do not reveal details about their classification model.

In conclusion, detection methods involving background subtraction or frame differencing are sensitive to illumination changes and shadows. Therefore, we select the histogram of oriented gradients to obtain accurate detection. We have found that detection performance in this constrained viewpoint is sufficient, whereas complex detection using CNNs[8] is considered too expensive in terms of computation. Given the previous work, we have adopted the AlexNet[10] network as the classification model and focus on an extensive evaluation of the large-scale MMR problem. As shown by Yang et al.,[12] AlexNet achieves state-of-the-art performance, is one of the fastest models at hand, and suitable for a real-time implementation.[17] Our experiments are performed on our proprietary dataset, which contains 10 times more images and the double amount of vehicle models than the public CompCar dataset,[12] but focuses on a single frontal vehicle viewpoint. We do not evaluate on the CompCar dataset because classification results are presented by Yang et al.[12] and we specifically aim at a large-scale evaluation.

3 System Description

The vehicle recognition system is shown in Fig. 2 and consists of two main components: detection and classification. The input of the detection component is a video stream from a camera mounted above the highway focusing on a single lane. The detection component localizes vehicles in each video frame. If a vehicle is found, the vehicle subregion is extracted from the video image. This cropped image is then processed by the classification component recognizing the make and model of the vehicle. During normal operation, all images from the camera are directly downsampled so that license plates are not readable anymore, while preserving sufficient resolution for classification. During training and validation, the original image resolution is used because the license plate information needs to be processed by an ANPR engine to automatically annotate the vehicle make and model label for our experiments (see Sec. 4). The detection and classification components are discussed below in more in detail.

3.1 *Detection: Vehicle Localization*

Vehicle detection is performed by sliding a detection window over the image and classifying each window location into object/background. A vehicle is detected when the image features at that location match with the classification model. The classification model is explained in more detail in the following paragraph. Sliding of the detection window over the image is performed at multiple, scaled versions of the input image and detections are merged by a mean-shift mode-finding merging algorithm. This detection process is repeated for every frame in the live video stream. Detections are tracked over time. For each vehicle, the subsequent make and model classification is performed once when the vehicle is fully visible in the camera view.

Since vehicle images contain large variations in appearance due to lighting, weather, vehicle type, and viewpoint variations, it is important to remove these variations by

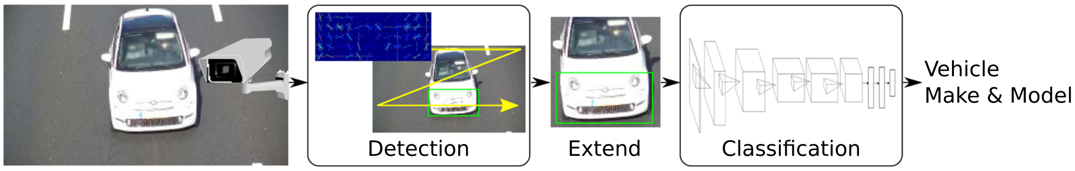

Fig. 2 System overview of the MMR system.

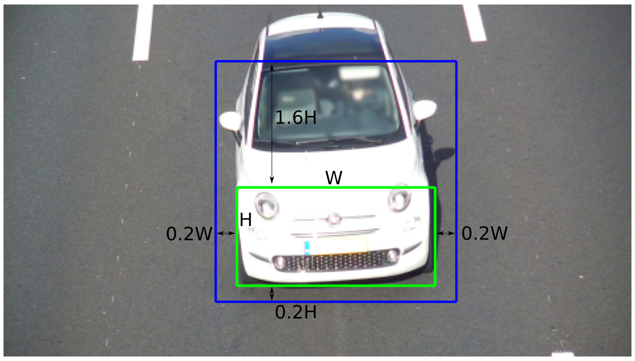

Fig. 3 Video frame, the detection box in green and classification ROI in blue. Windshield and license plate are blurred.

applying a feature transformation. We have selected the HOG feature transform, because of its high object detection performance and efficient computation. For each image pixel, the HOG features compute the local gradient magnitude and orientation. The gradient information is accumulated over small spatial subregions of the image (cells), and for each cell a histogram of gradient orientations is created. The histograms of the cells over an area in the image are concatenated and form the HOG feature description of that part of the image. The HOG feature description is then segmented into object/background using simple linear SVM classification. This simple classification model can be seen as a template that creates a description of the object in HOG feature space (example visualization in Fig. 2). More details about HOG can be found by Dalal and Triggs.[7]

In order to train the SVM, we use HOG features, which are computed for 12×5 cells of 4×4 pixels, covering the head lights and bumper of the vehicle (see green bounding box in Fig. 3 as an example). We use eight orientation bins ignoring the orientation sign with L2 normalization per cell. In addition to HOG features, the gradient magnitude for each cell is included in the feature vector. The linear classification model is trained from many vehicle and nonvehicle samples using stochastic gradient descent.[18] Vehicle tracking is implemented using optical-flow-based tracking of feature points using the concept of good features to track.[19]

3.2 *Classification: Make and Model Recognition*

Classification of make and model is performed once for each detected vehicle. The detection box is enlarged with a fixed factor to cover the grill, hood, and windshield, shown as the blue rectangle in Fig. 3. This part of the image is scaled to a fixed resolution of 256×256 pixels and used as the input to our MMR classifier in combination with the corresponding make and model class label.

We use the AlexNet classification model,[10] which is a CNN consisting of five convolution layers and two fully connected layers and a nonlinear operation between each layer. The output is a list of classification scores per vehicle model (class). The class with the highest classification score is the output of our MMR system.

The classification network is trained end-to-end on our vehicle images and class labels, which is then optimized to predict the correct vehicle class for each image. Note that we predict the make and model combination, so that the number of classes equals the number of vehicle models. We use the AlexNet network pretrained on ImageNet and fine-tune it on our dataset. For each training image, multiple random subimages of 227×227 pixels are used to train the CNN. We train for 50,000 iterations using a batch size of 128. All other training parameters are equal to the original model.[10]

4 Semiautomatic Dataset Generation

The detection and classification components both require offline training with vehicle samples prior to using them in the online MMR system. To train our recognition system, it would be trivial to collect raw video from the camera over a long period in the field and process this video data offline to prepare our training data. However, this approach cannot be followed because only a low-bandwidth connection exists between the roadside setup and the back-office, so that the amount of transferred video data is strictly limited. We propose a setup that consumes limited network bandwidth by only transmitting a selected single image for each vehicle that passes the camera. To implement this, a vehicle detector is required. We will now describe how the training data are collected for such a vehicle detector and the process of data collection for the classification component.

For both dataset collection purposes, we use an ANPR engine[20] that detects both the rectangular location of the plate and reads the characters (number). From the location of the number plate, we will create additional vehicle annotations to improve the detector, while from the recognized license-plate number, we look up the vehicle make and model from a database. Both procedures are visualized in Fig. 4. Next, we downscale each image to a lower resolution and only keep the make and model annotation while removing the license-plate number to anonymize the identity of the vehicle. With this data, we train our vehicle recognition system, which has a privacy-friendly design because there is no identity information and license plates are not readable.

4.1 *Training Data Collection for Detection*

We start by downloading a limited amount of video (15 min) and manually annotate 659 vehicles in these video frames. All annotations are flipped horizontally to obtain a total of 1318 annotated vehicles. Using these images, we train

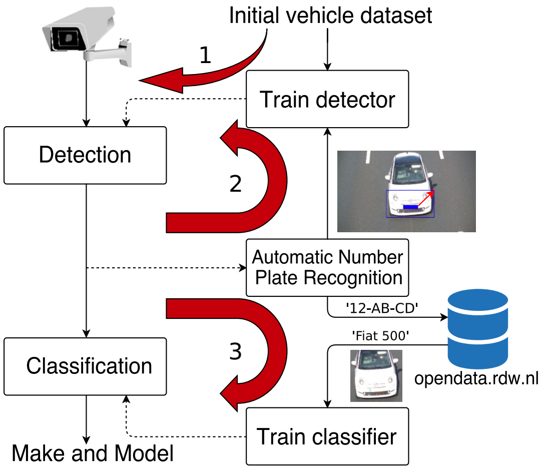

Fig. 4 Overview of the semiautomatic dataset generation procedure for the detection dataset (arrow 2) and our classification dataset (arrow 3) and the small initial vehicle detection dataset (arrow 1).

our initial vehicle detector and then apply this detector to the roadside setup, to collect images with vehicles and transmit these to our back-office. This approach is necessary because the detection performance of the initial detector is insufficient (resulting in missed cars and false detections). The initial detector is used at a low threshold to select all images that probably contain vehicles, see arrow 2 in Fig. 4. The bandwidth usage is limited by only downloading the selected images. We can now exploit these additional images to train an improved vehicle detector.

As manual annotation of vehicles is cumbersome, the downloaded images are annotated using an ANPR engine to locate the license plate. We assume that each vehicle has a license plate and use a fixed extension of the license plate box as a new vehicle annotation. If no license plate is found by the ANPR engine, we do not include the image in our dataset.

In total, we have collected 20,598 vehicle annotations during 4 h of online processing with our initial detector (including flipped versions). Half of this set is used to train the final detector, and half to evaluate the detector performance.

4.2 *Make and Model Attribute Acquisition for Classification*

The classification component requires sufficient samples for each vehicle class to distinguish intraclass variation from interclass variation. Moreover, not all vehicle models are equally popular, and the distribution of models is extremely nonuniform and unbalanced.

To collect data for training the classification component, we use our vehicle detector to automatically create cutouts of vehicles for a longer time period and send them to our back-office, see arrow 3 in Fig. 4. The network bandwidth is now limited by only transmitting cutouts of vehicles. Then, we process each vehicle cutout with an ANPR engine to find the license-plate number. The license-plate number is used to query a database with vehicle make and model information. In the Netherlands, such a database is provided and publicly available.[21] This process allows for large-scale annotation of our dataset.

The classification dataset was recorded during various weather conditions over a long interval of 34 days in which a total of 670,706 images (100%) were collected. Examples of dark and rainy samples and samples with strong shadows are shown in Fig. 5. All images are processed by the ANPR engine. In 649,955 of the images (97%), a license plate was found and the number could be extracted (other images contain too much noise for recognition). The make and model information was extracted from the database for 587,371 images (88%). Failure cases originate from non-Dutch license plates, which are not registered in the database and incorrectly read license-plate numbers (ANPR failure). In total, we detected 1504 different vehicle models. The distribution of the number of samples per vehicle model is shown in Fig. 6, which approximates a logarithmic behavior. The top-500 models all have more than 30 samples. The last 700 models only have one or two samples and represent various high-end vehicles, old-timers, and custom vehicles, such as modified recreational vehicles. The model that is mostly

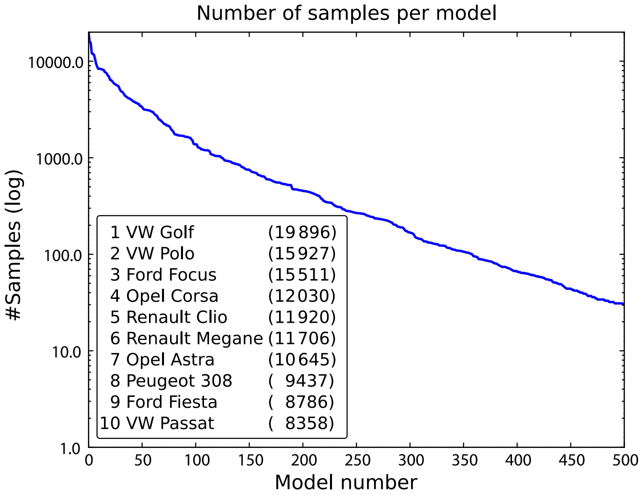

Fig. 6 Number of samples per model in our dataset.

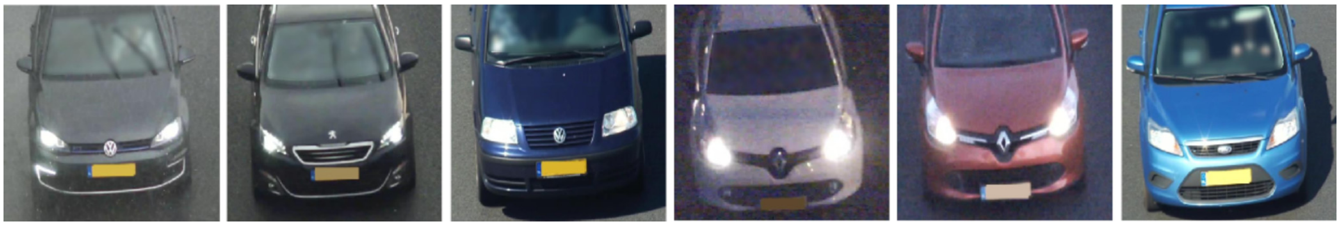

Fig. 5 Classification examples. Wind shield and license plates blurred for privacy.

detected is the Volkswagen Golf, with a total of 20k samples (13% of the dataset).

The classification dataset is split in a training set of 26 days (76%) and a test set of 6 days (18%), the remaining 2 days are used for validation during the training process to avoid overfitting on the training set. In total, we have created three different train, test, and validation datasets to enable a cross validation.

5 In-Depth Evaluation

This section evaluates the vehicle detector followed by an in-depth analysis of the make and model classification performance.

5.1 *Evaluation Metrics*

Detection performance is measured using recall and precision. A true positive (TP) rate is defined as a detection that has a minimum overlap (intersection over union) of 0.5 with the ground-truth box. Detections with lower overlap are false positives (FP). Missed ground-truth samples are denoted as false negatives (FN). The recall R and precision P are then computed as

$$R = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN}), \qquad P = \mathrm{TP}/(\mathrm{TP}+\mathrm{FP}). \tag{1}$$

We summarize the recall–precision curve by a single value as the area under curve (AUC), where perfect detection has a value of 100%.

The classification performance is measured by the top-1 accuracy, in which the number of correct classifications is divided by the total number of classifications performed. As a second metric, the performance per vehicle model is measured using recall and precision as in Eq. (1). A classification is a TP if the classification label is equal to the ground-truth label, otherwise it is an FP. A sample is an FN for a ground-truth vehicle model, if it is not correctly classified. Note that an FN for one class results into an FN for another class.

5.2 *Vehicle Detection*

This section evaluates the vehicle detection performance and compares the initial vehicle detector based on manual annotations with the final detector trained with the automatically collected vehicle annotations (Sec. 4.1). Figure 7 portrays the recall–precision curves for these detectors. The dashed blue curve shows the performance of our initial detector and the solid red curve depicts the results of the final detector. The initial detector already shows good performance, but regularly generates false detections. The final detector clearly outperforms the initial detector and is almost perfect with an AUC of 99%. The operation point has been empirically chosen to detect 98% of the vehicles, having negligible false detections, which is sufficient for the MMR application. Note that the reported detection rate holds for the per-image detection. Since a vehicle is visible in multiple video images, the actual vehicle detection rate is higher but is not measured this way.

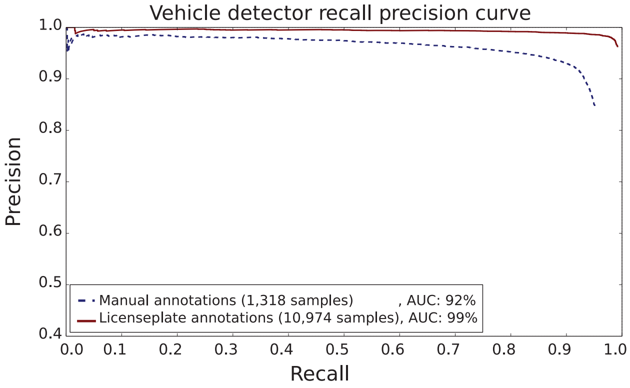

Fig. 7 Recall–precision curve of our initial and final vehicle detectors.

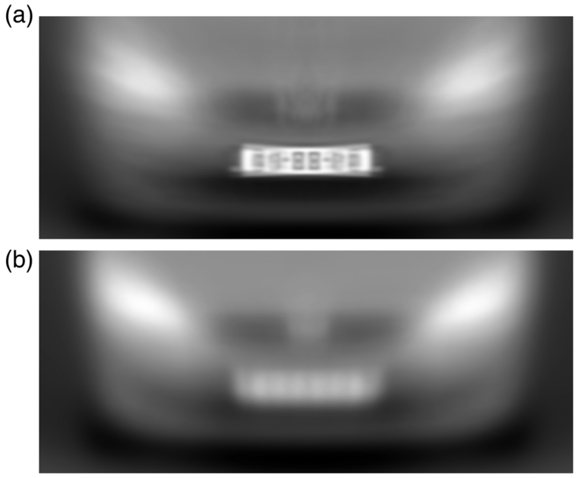

Fig. 8 (a) Average automatically annotated detection box and (b) average detected result.

In Fig. 8, the average images of our training set and our detector output are shown. The top image shows the average image of the annotations that are used to train the detector (the output of the ANPR detector). It can be clearly observed that the image is aligned on the license plate. The bottom image shows the actual detections after training. Note that the detector does not focus on the license plate only, but on the overall frontal-view layout of the vehicle. More specifically, the bottom view shows that all key elements of the frontal view are employed in a balanced way, as they appear at equal visibility. Apparently, those elements are learned and detected in a similar fashion. In conclusion, this highlights that our process of automatic annotation is quite powerful and the training process results in the generalization of the detector to the total vehicle characteristics.

5.3 *Make and Model Classification*

The classification performance is evaluated with three main experiments. First, we investigate the overall classification performance for an increasing number of vehicle models. For the vehicle models that are not considered in the training, we distinguish two cases: (1) we accumulate those into a single additional “other” class or (2) ignore them completely. Second, in a further experiment, we examine the performance per individual vehicle model in relation to the amount of samples per class. For the lower-performing models, we present a visual analysis. Third, we investigate the most informative part of the vehicle for classification by adding a synthetic occlusion to the vehicle images and measuring the effect on classification performance. Finally, we evaluate and compare other CNN architectures from literature on our vehicle classification problem.

5.3.1 *Overall performance*

Due to the nonlinear distribution of our make and model samples (classes) in our dataset (see Fig. 6), we investigate the classification performance when selecting an increasing number of classes in our model. One can simply ignore the samples of the nonselected classes or combine them into a single additional "other" class. This will enable the system to create awareness that it does not recognize these vehicles, instead of always misclassifying them. To compare the cases of having an other class or ignoring, two different models are trained. The first model ignores unconsidered classes completely (no other class) and the second model incorporates all unconsidered classes by explicitly adding an other class to our classification model. A simplified example of these two training methods is shown in Fig. 9, classifying between seven vehicle makes, two of these are considered other. We will now investigate these four combined cases: training and testing, both with and without the other class. The results of these cases are all shown in Fig. 10. Note that each combination of training and testing with/without is labeled as (a), (b), (c), and (d). We will now discuss each individual case.

Case (a). The classification accuracy of the model trained without the other class is constrained by the distribution of the data in the test set, e.g., for one class (VW Golf), the best possible accuracy is 13% because all samples in the other class will be wrongly classified (the "test all" case in Fig. 9). The results are shown in Fig. 10 by the solid blue line (a). The classification accuracy increases when more vehicle models are incorporated in the model and saturates around 500 classes, achieving an accuracy of almost 97%. This shows that the classification model is able to handle our large-scale classification task.

Case (b). Next, in case (b), we ignore the other class in our test set to measure the accuracy over the classes which are actually in our classification model. The results are shown by the dotted blue line (b) in Fig. 10. The performance gradually decreases when more vehicle classes are added to our classification model. This is as expected because the classification problem becomes harder when distinguishing more classes. In addition, with more classes, fewer samples are available for the additional classes (vehicle model #500 only contains 50 training samples).

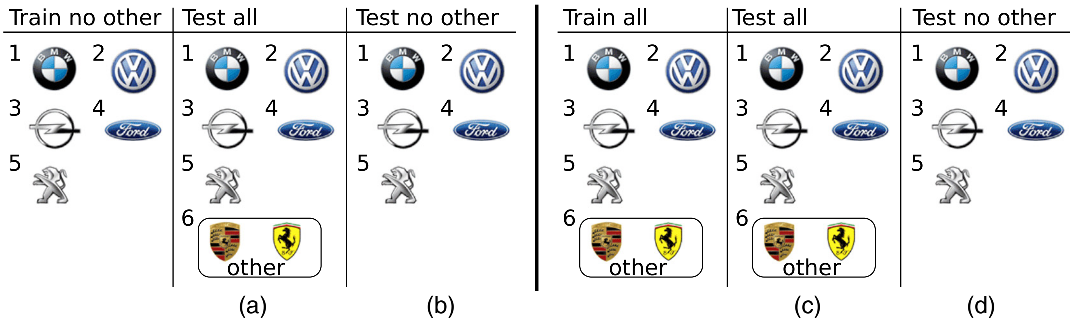

Fig. 9 Simplified example of training of the classification model with and without an explicit other class. At the left, training is performed without the other class and tested (a) with and (b) without other class. At the right, the other class is added during training and evaluated (c) with and (d) without this additional class.

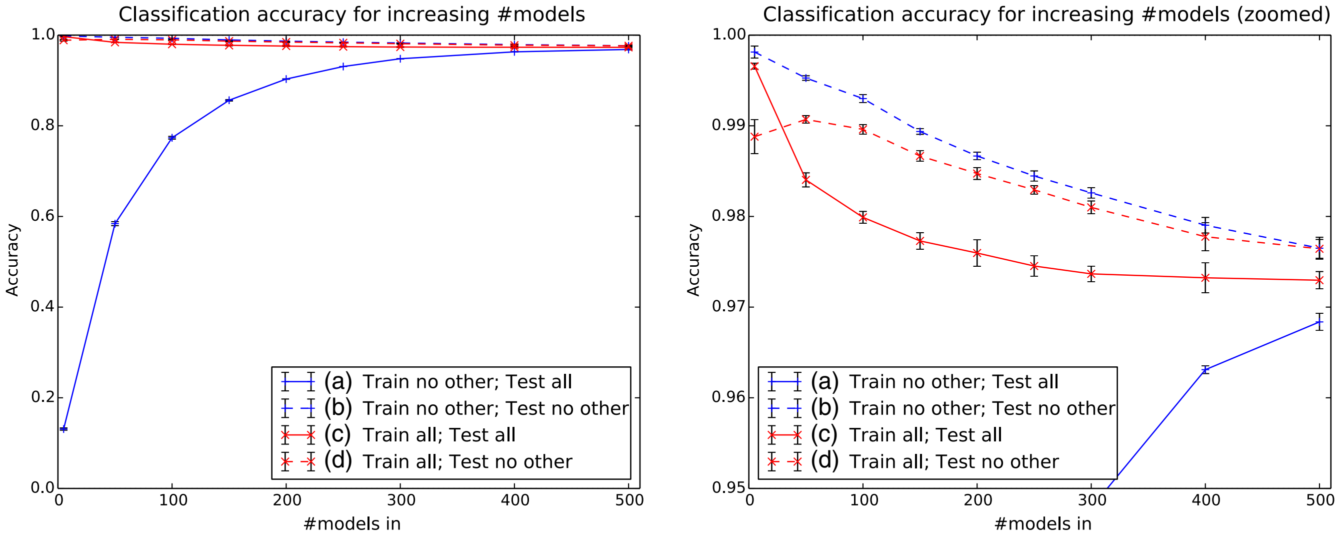

Fig. 10 Make and model classification accuracy for an increasing #models.

Case (c). Now we investigate case (c) and evaluate the effect of explicitly taking the "other" class into account in our classification model (train all in Fig. 9). This is interesting because when the other class is classified correctly, we can detect samples that the system is not aware of for future use. However, a lower accuracy is expected for the vehicle models which are incorporated in the classification model because the model has to deal with an extra class with a high amount of intraclass variation (it contains all other vehicle classes). The results are shown in Fig. 10 by the red solid line (c). Although the accuracy over the complete range is high (>97%), it continuously decreases for a growing number of classes. Over the total range of classes, the accuracy is 1% lower than the (b) model.

Case (d). When we evaluate this model over our test set without the other class [case (d)], we can measure the influence of this class on our classification performance and compare the results with the classification model that is not trained with the other class. The results are shown by the dashed red line (d) in Fig. 10. Note that the classification performance is a bit lower than the classification model trained without the other class, but it is approaching the performance for an increasing number of vehicle models. This is due to the decreasing number of samples in the other class while increasing the number of vehicle models in our classification model. Note that when training with all vehicle models in the classification model (without using the other class), there is no other class and all curves will have the same performance.

From this experiment, we can draw the following conclusions. The total performance for a number of classes will always be upper-bounded by case (b) and lower-bounded by case (a). When modeling more classes, the performance differences between the different cases become smaller. It is expected that the performance will converge at a large number of classes for all cases. However, because we only have a limited number of samples per class and the frequencies of occurrence become very small for the last number of classes, it is very difficult to experimentally validate this with sufficient data. Comparing cases (b) and (d), we can observe that when evaluating the influence of adding the other class to the training, this only marginally decreases the performances. As a bonus, it will become possible to exploit this additional class to extend our dataset. In the following experiment, we will validate this assumption by calculating the recall and precision for the individual vehicle classes.

5.3.2 *Per class evaluation*

In this experiment, we provide more insight into the classification performance per class. First, we evaluate the recall and precision for a different number of training samples per class to determine how many samples are needed to achieve good classification performance. Next, vehicle models with low accuracy are visually examined to determine the main cause of false classifications. Note that we fix the number of vehicle model classes to 500 (plus the additional other class) for this experiment. Each class has a different number of training samples due to the nonlinear vehicle model distribution. We measure the recall and precision for each class individually to evaluate the effect in classification performance per class. The results are shown in Fig. 11. Note that the plot is zoomed-in at sample sizes below 2000. For the 66 models having more than 2000 samples, recall and precision both approach unity (perfect classification).

It can be observed from the figure that for most classes with more than 500 training samples, the recall and precision exceed 95%. A notably lower performance is observed for classes with less than 200 samples. There are some outliers to this trend that perform significantly worse. The corresponding vehicle models are annotated in the figure. The other class has a recall of 74% and a precision of 84%, which is low compared to other classes with many samples. However, this can be explained by the large intraclass variation in this class compared to the normal vehicle classes. Nevertheless, the classification model is able to detect vehicle models that are not present in our training dataset. When using this other class to collect additional training samples for classes that are not occurring at all in the training set, a high precision results in effectively selecting these samples. For every 100 images automatically classified as other, 84 are actually useful. Moreover, we have found that of the 16% that is misclassified as other, the make is typically

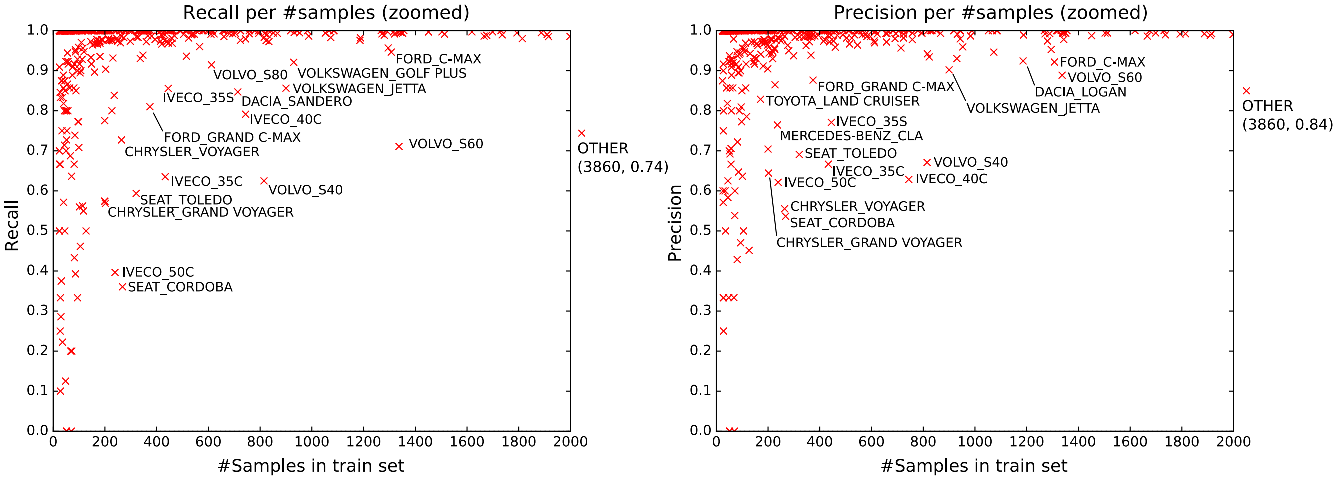

Fig. 11 Recall and precision for the amount of training samples available.

correctly classified and the model differs only slightly from the ground-truth. Many of these cases result from incorrect labeling in the constructed dataset, both for training and testing (e.g., Citroen DF instead of Citroen DS and Saab 93 versus Saab 9-3).

Visual examples of outliers are further investigated in Fig. 12. This figure illustrates an example TP classification and the highest FP classifications. We observe that for these cases, either the class labels are inconsistent (for example, Citroen DS3 and DS 3) or the classes are visually similar. For example, the Iveco model number relates to the wheel base and payload capacity, which cannot be visually observed from the front of the vehicle. Other difficult cases are the more visually similar sedan models versus estate versions of a vehicle model (Volvo S40 versus Volvo V50).

Note that for the Volvo S40, there is an example of a ground-truth label Renault Twingo which is incorrectly annotated. An empirical evaluation of all FP classifications shows that about 0.18% of the samples in the test dataset have incorrect labels (this is an estimate because the authors are not car experts). These incorrect labels are caused by the ANPR engine, resulting in an incorrect license-plate number due to lighting or dirt on the plates. These incorrectly read license plates can actually correspond to registered vehicles in the online registration database, which finally lead to labeling errors in the dataset.

5.3.3 *Most informative region*

The classification model uses the total vehicle image as input for the classification. In this experiment, we will investigate which vehicle region is most important and informative for classification. If an image region is assumed to contain no information for make and model classification, we could potentially exclude that region in our classification model. Vehicles are left/right symmetrical giving a vertical symmetry axis, so we expect the classification performance not to drop when half of the vehicle is occluded.

We measure which region is most important for classification with two experiments. In the first experiment, the accuracy is measured when increasingly occluding the complete test set from zero occlusion to full occlusion. This is performed in four different directions: from left to right, top to bottom, right to left, and bottom to top. The results are shown in Fig. 13. It can be observed that the accuracy for occlusion from left to right and right to left are similar, both have a high accuracy until 50% of the area is occluded. This shows that our classification model can handle large occlusions from both sides and that there is an equal amount of vehicle model information at both sides of the image, confirming the symmetry of vehicles. When occluding the vehicles from top to bottom, the accuracy drops after 25% occlusion, meaning that vehicle model information is contained within the windscreen. Occlusion from bottom to top results in a significant decrease in accuracy above 25% occlusion. This point corresponds to the vehicle grill (see the bottom row in Fig. 13 left at 25% occlusion). We can conclude that most information is contained in the bottom half of the vehicle.

After evaluating an increasing amount of occlusion (from zero to full occlusion), we now measure the effect of occlusion by sliding an occlusion patch with fixed size over the image. This allows to measure the drop in classification performance when covering specific regions of the vehicle. The classification top-1 score of all vehicles in the dataset is measured for each position of the occluding patch. For a single image, the score is accumulated only if the vehicle is

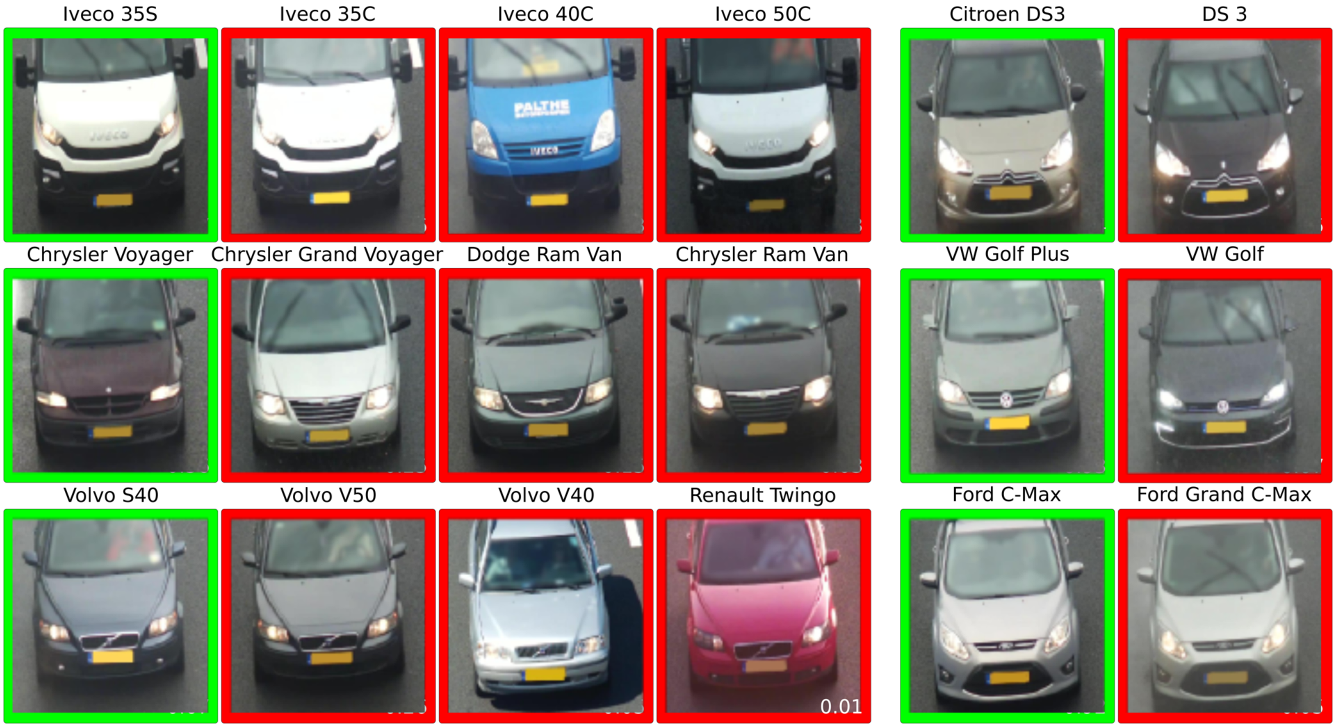

Fig. 12 TP classification (green) and the strongest FP classifications (red) for several models with low precision. The number at the bottom-right of TP (green) represents the precision, the number of FP (red) denotes the false positive rate.

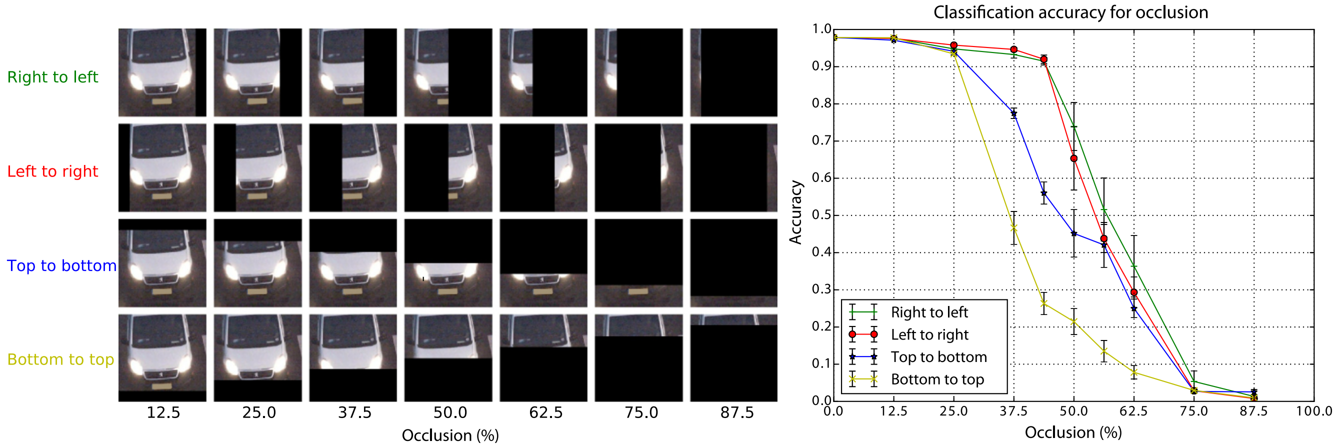

Fig. 13 Artificial occlusions added to the dataset and the effect on classification performance.

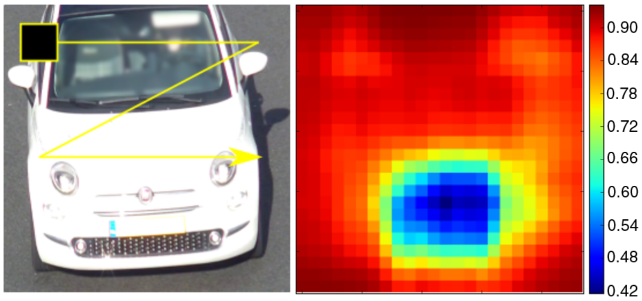

Fig. 14 Effect of occlusion on classification using an occlusion window with fixed size.

classified correctly and made zero otherwise. For each patch location, we normalize the score to the total number of images.

This approach is performed by sliding a window of size 64 × 64 pixels over the image with a step size of 8 pixels. The sliding is performed over the complete test set, as we have already localized the vehicle with our vehicle detector and we assume that the most important region is similar for all vehicles. The resulting heatmap is shown in Fig. 14, where red means high score and blue means low score in classification. Since the occlusion removes information, the blue region denotes the most informative region because the classification score is lowest when this region is occluded. This region covers the grill of the car, typically also containing the brand logo. Regions that have a small but notable influence are the headlights and the upper corners of the windshield.

5.3.4 *Comparison with other classification MODELS*

In our final experiment, we compare the AlexNet[10] classification model with other models from the literature: ResNet-50,[22] VGG16,[23] and SqueezeNet V1.1.[24] The models are trained with their default parameters and fine-tuned, using their pretrained models from the ImageNet classification competition. ResNet-50 is fine-tuned for 200,000 iterations with batch size 32 on random crops of 224 × 224 pixels taken from our input samples of 256 × 256 pixels. VGG16 is fine-tuned with batch size 64 for 100,000 iterations, with random 224 × 224 pixel crops. SqueezeNet uses random 227 × 227 pixel crops and is fine-tuned for 200,000 iterations with batch size 32. In addition, mirrored versions of the input samples are used for data augmentation. All other parameters are similar to the original implementations. Training is performed three times for 500 models and the other class. The results are shown in Table 1. All classification models achieve an accuracy of 97% or higher, which shows that all models can handle the large-scale make and model classification task.

Table 1 Comparison with other classification models for 500 vehicle classes and “other” class.

CNN model	Accuracy	Std. dev.
AlexNet[10]	97.29	0.08
ResNet-50[22]	97.27	0.06
VGG 16[23]	97.92	0.03
SqeezeNet V1.1[24]	97.93	0.14

Because performance differences are small, we can conclude that the classification problem can be effectively solved by any of these CNN models. This suggests that the experimental validation is a general evaluation of the dataset and is not dominantly influenced by the combination of the dataset and applied classification model. It may be possible to combine the outputs of the CNN networks to possibly obtain a higher accuracy. This is left for future work, since it would clearly increase the complexity and complicate our real-time requirements.

6 Application

In this section, we evaluate the performance of the complete system as described in Sec. 3. The extracted make and model information is used to assist the police in a law-enforcement application where stolen vehicles are recognized. In order to measure the system performance for this application, a field test has been conducted by an external party using live video from a camera mounted above a highway in the Netherlands.

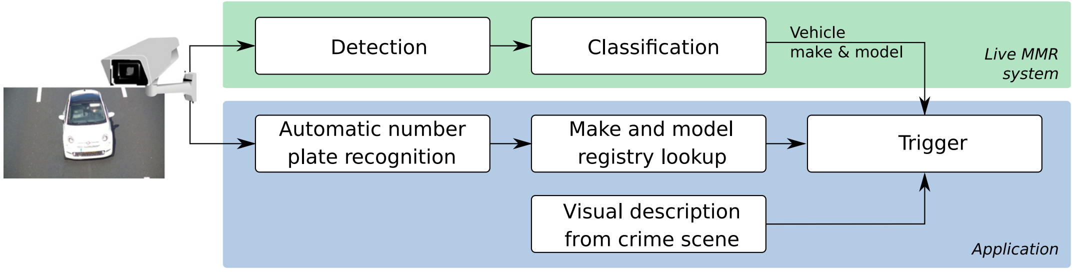
Detection
Classification
Vehicle make & model
Live MMR system
Automatic number plate recognition
Make and model registry lookup
Trigger
Visual description from crime scene
Application

Fig. 15 The final application in which our MMR system is used.

The make and model information provided by the visual recognition system is used in two ways, see Fig. 15. The first application is to continuously monitor the highway and compare the visual MMR results with the make and model information obtained by querying the vehicle registry with the license plate. A mismatch between these results indicates possible license plate fraud or a missing license plate. The second application is focused on localizing vehicles involved in criminal activities without requiring knowledge on the license plate of the respective vehicle. Using only a witness description of the vehicle model, the police can now actively search if the vehicle passes the camera system. Currently this is only possible if the license plate is *a priori* known. By using the visual information for MMR recognition, it is more difficult for criminals to circumvent this localization by replacing or removing the license plate. The evaluation of the complete system performance is measured using the results of an ANPR engine as ground truth. The ANPR engine detects the license plate and uses the plate number to query the make and model information in a vehicle registration database (similar to Sec. 4). Manual verification of the external party is then carried out to verify the ANPR engine and evaluate the results of our system. Note that when evaluating the complete system performance, a vehicle that is not detected (and thus not classified) will result in a misclassification. Hence, combining the performance from our experiments for detection (98.00% of the vehicles) and classification (97.29%) leads to an expected total system accuracy of 95.34%. In the live system, we deploy the vehicle detector from Sec. 5.2 and our classification model trained on all data incorporating the other class (see Sec. 5.3.1). In the following, we first present the computation time of the individual recognition components and then discuss the field test.

6.1 *Real-Time Computation Performance*

Detection and tracking is performed every frame in the live video, while classification is performed only once per vehicle. To ensure real-time processing, the MMR classification stage is implemented in a separate thread. The recognition system is mapped on low-cost computing hardware consisting of a dual-core CPU (i3-4170 at 3.70 GHz) with 2-GB RAM memory. The real-time performance is measured by processing 15 min of video (at 30 frames per second) on the dedicated hardware platform. The evaluation has been performed during rush hour to measure the performance in real-life practical situations with the highest traffic intensity. During the benchmark, we measure the number of function calls to each stage and their average computation time. Note that detection and tracking are applied every video frame to localize vehicles. Classification is only applied once for every detected vehicle. The video capture time is measured separately and the overhead in threading, visualizing, and storing the results is measured together as other. The results are shown in Table 2.

Table 2 Timing characteristics for evaluating the computational performance.

Function	Calls	Avg. time (ms)	Avg. time per frame (ms)
Detection	27,000	0.82	0.82
Tracking	27,000	2.77	2.77
Classification	493	178.10	3.42
Video capture	27,000	2.31	2.31
Other	—	—	1.06
Total	—	—	10.38

Detection and tracking takes only 3.59 ms/frame, while classification consumes 178.10 ms per vehicle. For typical rush-hour traffic, the system executes with real-time performance. All stages take an average of 10.38 ms/frame of the total budget of 33.33 ms/frame. To further check the real-time system operation, we now calculate the performance under a worst-case situation, which occurs when two consecutive vehicles drive very close behind each other. In this scenario, we assume vehicles of 4 m length, driving 200 ms apart with a speed of 130 km/h, which is an extreme form of tailgating. This leads to a classification every 310.77 ms. This relates to 10.36 frames for detection, tracking, video capture, and other calls with a total time budget of 72.10 ms. The total computation time then amounts to 250.20 ms, which is within our 310.77 ms budget. This shows that the hardware setup (without GPU acceleration) is sufficient to perform real-time MMR, even in worst-case scenarios.

6.2 *Field Test*

The proposed MMR system has been deployed as a live system for the Dutch National Police in the Netherlands. An independent evaluation of our system has been carried out by an external party. The third party used the same camera stream for our visual MMR system and connected it to an ANPR engine. To validate the visual MMR system, the ANPR results were manually compared to our system

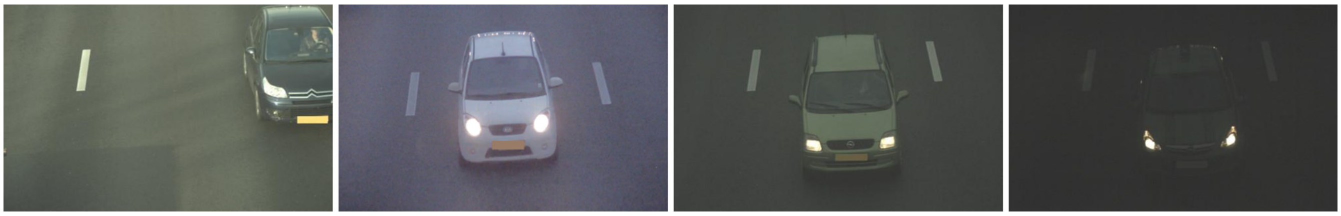

Fig. 16 Example frames taken during the field test in various light conditions.

output. This evaluation uses our top-500 classification model incorporating the other class and processed four different time periods with a total duration of 8 h, under different weather and lighting conditions (see Fig. 16 for some example frames captured during the field test).

The external party reported an overall accuracy of 92.4%. When only the make classification is considered, an accuracy of 95.7% was measured, indicating that the make is classified correctly, but the model was erroneously classified for 3.3% of the vehicles. Visual interpretation of the results indicates that errors mainly occur due to partly visible vehicles. These vehicles are not detected by our system and therefore not classified (thereby lowering the accuracy). The ANPR-based validation process locates the license plate in the image and therefore does produce a classification. Other errors occur from low-light conditions, where vehicles are barely visible. These errors explain the small performance gap between the results from the field test (92.4%) and the overall system accuracy from our own benchmarks (95.34%).

7 Conclusions

We have proposed a system for vehicle MMR that automatically detects and classifies the make and model of each vehicle from a live camera mounted above the road. We have shown that with minimal manual annotation effort, we can train an accurate vehicle detector (99% AUC), by using an automatic number plate recognition (ANPR) engine. During testing, the ANPR testing is not required anymore to obtain a high detection performance. The applied vehicle detector automatically detects vehicles and by additionally extracting the license-plate number, the make and model information is obtained from a database.

For classification, we have used a CNN and have experimented with the AlexNet model, leading to an MMR classifier with a top-1 accuracy of 97.6% for 500 vehicle models. An explicit other model class only leads to a small drop in performance (∼0.3%), but makes the model aware of unrecognizable vehicles. This approach can be used to automatically gather additional samples of rare and new vehicle models to further improve the classification model.

We have evaluated the effect of the number of training samples per class and conclude that the classification performance is high when more than 500 samples are available. The performance significantly drops at 200 samples per class or lower. A visual inspection of low-performance classes reveals that the problem is ill-posed. Some vehicle models are defined by properties that cannot be visually distinguished from the vehicle front, such as the difference between sedan and estate models or engine details. These models should be joined in a combined model description, or additional input data (e.g., a side view) are required to solve such detailed classification tasks. Other notable errors occur from inconsistent model label definitions or incorrect labels, resulting from misreadings in the ANPR engine, errors in the national vehicle database or false license plates.

To investigate the most informative vehicle regions, we have imposed occlusions at various visual positions and then measured the effect on the classification performance. We have shown that the bottom of the vehicle is most informative, although the windshield region also contains information. Since vehicles are symmetrical giving a vertical symmetry axis, the performance is not significantly penalized when only the left or right of the vehicle is visible. By sliding a smaller fixed-size occlusion region over the images, we have shown that the grill and brand logo are most informative for classification.

The evaluation of different state-of-the-art CNN models reveals that the resulting classification performance is similar for all CNN models. This implies that the experimental validation is a general evaluation of the dataset in combination with a state-of-the-art CNN, where the choice of the CNN model is less relevant.

The proposed semiautomatic system can be used to effectively construct a large dataset, which in turn can be applied to train an accurate recognition system. The detailed investigation of this paper shows that most classification failures originate from errors in the dataset in which the estimated class was correct. To fix the errors in the dataset, and thereby improve the accuracy of the classification model, manual inspection of the failure cases is required. These failure cases are ill-defined vehicle models that are not visually distinguishable and incorrectly labeled samples. Despite the occurrence of such vehicle models, the classification model is able to cope with this noise in the dataset and can accurately recognize vehicle models. It was shown that the system executes in real-time performance without GPU support. To achieve this, the classification stage was implemented in parallel. The system was successfully applied in a field test with the Dutch National Police involving four intervals of 8 h, yielding an overall accuracy of 92.4%.

Disclosures

The Dutch National Police has facilitated the experiments from this study but without further financial support. This research has been funded by a national research Impulse program.

References

1. M. H. Zwemer et al., "Semi-automatic training of a vehicle make and model recognition system," in *Proc. of 19th Int. Conf. on Image Analysis and Processing (ICIAP 2017), Part II, Catania, Italy*, S. Battiato et al., Eds., pp. 321–332, Springer International Publishing, Cham (2017).

2. Y. Ren and S. Lan, "Vehicle make and model recognition based on convolutional neural networks," in *7th IEEE Int. Conf. on Software Engineering and Service Science (ICSESS)*, pp. 692–695 (2016).
3. J. Prokaj and G. Medioni, "3-D model based vehicle recognition," in *Workshop on Applications of Computer Vision (WACV)*, pp. 1–7 (2009).
4. A. J. Siddiqui, A. Mammeri, and A. Boukerche, "Real-time vehicle make and model recognition based on a bag of surf features," *IEEE Trans. Intell. Transp. Syst.* **17**, 3205–3219 (2016).
5. V. S. Petrovic and T. F. Cootes, "Analysis of features for rigid structure vehicle type recognition," in *Proc. of the British Machine Vision Conf.*, Vol. 2, pp. 587–596 (2004).
6. R. G. J. Wijnhoven and P. H. N. de With, "Unsupervised sub-categorization for object detection: finding cars from a driving vehicle," in *IEEE Int. Conf. on Computer Vision Workshops (ICCV Workshops)*, pp. 2077–2083 (2011).
7. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1, pp. 886–893 (2005).
8. Y. Zhou et al., "DAVE: a unified framework for fast vehicle detection and annotation," *Lect. Notes Comput. Sci.* **9906**, 278–293 (2016).
9. Y. LeCun et al., "Backpropagation applied to handwritten zip code recognition," *Neural Comput.* **1**(4), 541–551 (1989).
10. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (2012).
11. O. Russakovsky et al., "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vision* **115**(3), 211–252 (2015).
12. L. Yang et al., "A large-scale car dataset for fine-grained categorization and verification," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Boston, Massachusetts, pp. 3973–3981 (2015).
13. P. Sermanet et al., "Overfeat: integrated recognition, localization and detection using convolutional networks," in *Int. Conf. on Learning Representations (ICLR)*, Banff, Canada (April 2014).
14. C. Szegedy et al., "Going deeper with convolutions," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2015).
15. J. Sochor, A. Herout, and J. Havel, "Boxcars: 3-D boxes as CNN input for improved fine-grained vehicle recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 3006–3015 (2016).
16. A. Dehghan et al., "View independent vehicle make, model and color recognition using convolutional neural network," in *CoRR*, arXiv:abs/1702.01721 (2017).
17. A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," in *CoRR*, arXiv:abs/1605.07678 (2016).
18. R. G. J. Wijnhoven and P. H. N. de With, "Fast training of object detection using stochastic gradient descent," in *Proc. of Int. Conf. on Pattern Recognition*, pp. 424–427, IEEE Computer Society, Washington, DC (2010).
19. J. Shi and C. Tomasi, "Good features to track," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 593–600 (1994).
20. ARH, CARMEN FreeFlow ANPR/LPR Software, 2016, http://www.arhungary.hu/ (May 2017).
21. RDW: Dienst Wegverkeer (Dutch agency for road transport), Open mobiliteitsdata, 2016, https://opendata.rdw.nl/ (May 2017).
22. K. He et al., "Deep residual learning for image recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, Nevada, pp. 770–778 (2016).
23. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. on Learning Representations (ICLR)*, San Diego, California, (May 2015).
24. F. N. Iandola et al., "Squeezenet: alexnet-level accuracy with 50x fewer parameters and <1 mb model size," in *CoRR*, arXiv:abs/1602.07360 (2016).

Matthijs H. Zwemer received his BE degree in electronics and technical information technology from HZ University of Applied Sciences, The Netherlands, in 2008. In 2014, he graduated in electrical engineering from Eindhoven University of Technology, The Netherlands, and joined ViNotion, where he works on object detection and classification for traffic analysis in computer vision. Since 2015, he is a PhD candidate in the video coding and architectures group at Eindhoven University of Technology, The Netherlands.

Guido M. Y. E. Brouwers received his MSc degree in electrical engineering at Eindhoven University of Technology, The Netherlands, in 2015. For his master's thesis, he focused on automatic camera calibration for surveillance cameras. After graduating, he joined the ViNotion research team. He has worked on object detection, tracking, and classification in various application fields. These application fields include crowd analysis, traffic analysis, and maritime analysis.

Rob G. J. Wijnhoven graduated in electrical engineering from the Technical University Eindhoven in 2004. From 2004 to 2009, he worked on object categorization for video surveillance at Bosch Security Systems, Eindhoven, The Netherlands. In 2009, he joined ViNotion, where he is currently responsible for the research in computer vision. In 2013, he obtained his PhD in object categorization and detection. His interests include object detection, tracking and classification, and their integration in industrial applications.

Peter H. N. de With (fellow IEEE) received his PhD from the University of Technology Delft, The Netherlands. In 1984 to 1997, he was senior TV Systems Architect at Philips Research Eindhoven. He was a full professor at the University of Mannheim, Germany, from 1997 to 2000. In 2000 to 2007, he was consultant at LogicaCMG in Eindhoven and part-time professor at the Eindhoven University of Technology. In 2008 to 2010, he was VP at Cyclomedia. Since 2011, he is a full professor at Eindhoven University of Technology.