Accurate classification for Automatic Vehicle Type Recognition based on ensemble classifiers

Nadiya Shvai[a], Abul Hasnat[a], Antoine Meicler[a], and Amir Nakib[a,b]
[a]*Cyclope Team, VINCI Autoroutes, Rueil Malmaison, France.*
[a,b]*University Paris Est, Laboratoire LISSI, Vitry sur Seine, France.*

***Abstract*—In this work, a real world problem of the vehicle type classification for Automatic Toll Collection (ATC) is considered. This problem is very challenging because any loss of accuracy even of the order of 1% quickly turns into a significant economic loss. To deal with such problem, many companies currently use Optical Sensors (OS) and human observers to correct the classification errors. Herein, a novel vehicle classification method is proposed. It consists in regularizing the problem using one camera to obtain vehicle class probabilities using a set of Convolutional Neural Networks (CNN), then, uses the Gradient Boosting based classifier to fuse the continuous class probabilities with the discrete class labels obtained from OS. The method is evaluated on a real world dataset collected from the toll collection points of the VINCI Autoroutes French network. Results show that it performs significantly better than the existing ATC system and, hence will vastly reduce the workload of human operators.**

***Index Terms*—Vehicle Classification, Convolutional Neural Network, Gradient Boosting.**

I. Introduction

Automatic Vehicle Recognition (AVR) is an important prerequisite for a number of different domains, such as smart city, autonomous vehicle, intelligent transportation, traffic analysis, vehicle security, ATC, Vehicle Model and Make Recognition (VMMR), etc. [1]–[8]. Recent progress in high performance computing, such as cloud computing with graphics processing units (GPU), coupled with the advances of machine learning algorithms, such as CNN [9], [10], accelerates the development of a number of computer vision based solutions for AVR [2], [4]–[6], [11]–[21].

ATC is a practical use case of AVR[1]. Although the ATC systems are already deployed in many countries [7], human efforts are yet very much necessary to manually correct the misclassifications because of economic consequence of any loss of the classification system. This research is motivated from this and aims to improve an existing ATC system that uses OS to classify vehicles. However, the OS makes misclassifications due several reasons, such as sensor noise, measurements proximity of inter-class vehicles and additionally attached items with vehicles. While an obvious solution is to update the OS itself, an alternative is to exploit additional sources of information. Existing ATC setup (see Sect. III) captures images for the human operators to manually correct the misclassifications. This opens the opportunity to exploit the images and develop a computer vision based vehicle type classifier using an efficient method, e.g., the CNN [23]. Moreover, this classifier can be integrated with the OS to further enhance the overall performance. The above points motivate to develop a novel system which exploits information from both OS and camera by combining results from the OS, and image classifiers.

To develop an efficient ATC system, a challenging dataset is collected, where vehicles are categorized[2] based on their physical measures that decreases the inter-class and increases the intra-class variations. Sect. III provides the details specification for each class/category of vehicles. Besides, it comprises a large variations of the captured images with different conditions, *e.g.,* illumination, occlusion, poses, localization, multi-vehicle presence, etc., for instance see Fig. 3. This enhances the visual intra-class variations (*e.g.,* see Fig. 1). and makes the vehicle classification a significantly challenging problem.

Fig. 1: Appearance variations of the heavy vehicles from the same category due to pose, perspective, lighting and occlusion.

CNN based [23] methods become the *de facto standard* for image-based object recognition [24]. It has been vastly adopted by the recent image-based[3] vehicle classifiers [2], [4]–[6], [17]–[20]. Most of them applies the vehicle *detection followed by classification* approach, which has several drawbacks: (a) dependency on the detectors' performance; (b) hard to determine the true category when multiple vehicles are present, see Fig. 2 and (c) increase of the computation time. Considering these, this research uses the *detection-free and holistic-scene* based approach for the *CNN based* classifiers.

[1]The classification goal and challenges of AVR depend on the target tasks, *e.g.,* verification [1], [21], ATC [7], [22] and VMMR [4], [19]. While verification measures the similarity of two vehicle images, VMMR performs fine grained classification with hundreds of possible classes. The ATC task of this research can be distinguished from them based on the different specifications for vehicle categorization provided by the company. Moreover, the ATC specifications can be different for different countries [7].

[2]This is unlikely to the existing datasets, *e.g.*, CompCars [19] for VMMR, where the vehicles of are categorized based on their models and makers.

[3]Vehicle classification has been performed by different sources of information, where *image* is one of the important source (see Sect. II for others).

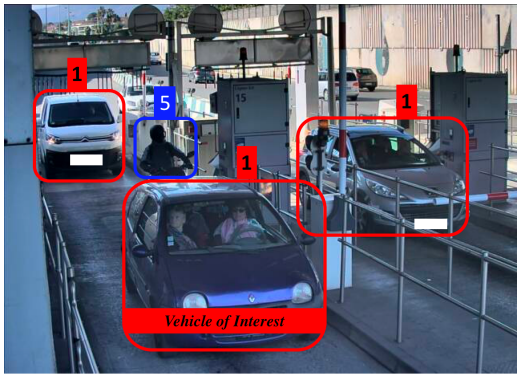

Fig. 2: Illustration of the multi-vehicle scenario.

The proposed *CNN based* classifiers achieves significant improvement as a stand-alone classifier. However it has some limitations, e.g., from the frontal view images it is difficult to distinguish the categories of heavy vehicles, which can be well classified with the OS. This indicates that a robust method for the ATC problem can be developed by efficiently combining the image based classifiers with the OS decisions. Therefore, a novel vehicle type recognition method is proposed, which performs the ensemble approach at two stages/layers:

1) ***1st layer:*** combines the outputs from two different types of classifiers: (a) OS based and (b) CNN [9], [10] based. It provides a concatenated vector (of decision and uncertainty) as an input to the next ensemble layer.
2) ***2nd layer:*** combines the classifiers decisions obtained by training on different weighted sets of the data (output from *feature layer*). The Gradient Boosting [25], [26] (GB) method is applied to perform this task.

The proposed method is evaluated on the collected dataset and compared with the existing system. Results indicate that it significantly outperforms the existing system with a large (99.03% compared to 52.77%) margin and hence alleviates the necessity to employ the vast amount of human efforts. Besides, comparison with a set of CNN based methods shows that it performs better than the *state-of-the-art* approaches.

The contributions of this research can be summarized as follows: (a) introduce a challenging and practical *vehicle classification* use case in the context of ATC; (b) propose a novel vehicle classification method; (c) propose a modification of VGG-16 CNN architecture which significantly ($\approx$ 9 times) reduces its complexity; (d) achieved very high accuracy and significantly outperform the existing system (e) provide interesting visual explanations about a CNN classifier from both *vision* (using GradCam [27]) and *learning* (using t-SNE [28]) perspectives and (f) provide an in-depth analysis to discover the remaining challenges and explore the future works.

This work extends our recent work [29] by incorporating: (a) additional study of the related work; (b) newer contributions on the method by changing from *CNN+OS* to *CNN1+CNN2+OS*; (c) extensive experiments and evaluation with possible alternatives methods from the *state-of-the-art* and (d) enhanced discussions with different visualization strategies.

The outline of the rest of this paper is as follows: Section II discusses the related works. Section III presents the formulation of the problem and describes the vehicle classification dataset. Section IV presents the proposed method. Section V provides experimental results and discussion. Finally, Section VI draws conclusions and discusses future perspectives.

II. Related Work

Multi-sensor and multi-modal data fusion [30] is a well-known technique applied to different (supervised [3] and unsupervised [31]) classification problems based on two main strategies - early fusion [31] and late fusion [3]. The late fusion [3] approach fuses the decisions from multiple classifiers used to classify different modalities/sensors data. *This research exploits the late fusion-based strategy for the proposed method.*

Multi-Classifier Systems [32], [33] is a popular learning-based decision fusion method to combine the classifiers from different backgrounds. *The proposed method follows the parallel ensemble design topology (at the 1st-layer) and employs a trainable fusion approach (Boosting [25]) which diversifies the input data distribution for each of its individual classifiers (at the 2nd-layer)*. Moreover, following the cascaded classification model [34], this work uses multiple CNN classifiers.

Vehicle classification is often accomplished with different sensors, such as strain gauge [35], Light Detection And Ranging (LiDAR) [3], [36] and cameras [3], [5], [36]. Numerous methods [3], [36] combined these sensors to enhance accuracy. The late fusion strategy is more popular [3], [35], [36], where the classification decisions are fused by different methods, such as the Support Vector Machine (SVM), Neural Networks (NN), Random Forest (RF), Gradient Boosting (GB), etc. [37]. *This research uses the GB method to combine the classifiers*.

Recently Oh and Kang [36] proposed a fusion method to recognize 3-classes (car, pedestrian and cyclist). They used the late fusion strategy with the SVM classifier to combine the features from the independent CNN models trained with the CCD and LIDAR sensors data. Concurrently, Asvadi *et. al.* [3] proposed a late-fusion based vehicle detection method with different modalities of data, such as color image, dense depth-map and reflectance map. They employed a multi-layer Perceptron Neural Network [37] to fuse the object bounding boxes obtained from the independent CNN models trained with each modality. *Despite the similarity to the fundamental notion of these methods, the approach of this research is different, because: (a) it performs classification rather than detection; (b) the modalities are different (camera and OS) and (c) it uses a different and challenging dataset.*

Most of the *single sensor* based vehicle classification methods target the application of VMMR. These methods commonly apply vehicle detection before classification. They can be categorized into *non-CNN-based* [11]–[15] and *CNN-based* [2], [4]–[6], [17]–[20]. The *non-CNN-based* approaches employed different keypoint and region based descriptors to extract detected vehicle features and then applied different learning methods to classify the vehicles. Dlagnekov and Belongie [11] used the Scale Invariant Feature Transform (SIFT) keypoints and descriptors with the keypoints matching technique. Pearce and Pears [12] used the Harris corner strengths with a Naive Bayes Classifier. Jun-Wei *et. al.* [13]

used the Histogram of Oriented Gradients (HOG) and Speeded Up Robust Features (SURF) descriptors with the SVM classifier. Siddiqui *et. al.* [15] used the Bag-of-SURF features with the SVM classifier. He *et. al.* [14] extracted normalized illumination and texture features with the RUSBoost based ensemble classifier. Zhang [38] used Gabor Wavelet and Pyramid HOG and proposed a cascaded classifiers ensemble with k-nearest neighbors, Multilayer Perceptron (MLP), SVM, RF and Rotation Forest. *The proposed method is different than these methods due to the target application, i.e., ATC, use of multiple sensors information, use of CNN models, use of the detection-free vehicle classification approach and use of a different learning method for the ensemble.*

Recently the *CNN-based* VMMR (also called *fine grained vehicle classification*) becomes very popular due to the breakthrough performance from the CNN-based object classification methods [23], [24] and the publicly available vehicle datasets [19], [21]. Biglari *et. al.* [4] proposed a cascaded part-based system, which employs the latent SVM method with the part-based CNN models. Similarly, Fang *et. al.* [17] applied the SVM classifier on the concatenated features extracted from multiple CNNs. Their CNN models learn to classify a vehicle based on its holistic (coarse) and parts (fine) images. Unfortunately, both approaches [4], [17] are sensitive to viewpoints and degrade performance for the non-frontal views of vehicle images. Hu *et. al.* [20] proposed a spatially weighted pooling with different CNN models to extract important features from the image. *Unlike these, the proposed method is not sensitive to vehicle viewpoints. Moreover, it considers the holistic image as input rather than the cropped vehicle image.*

Other than the part-based approaches, Wang *et. al.* [6] applied deep learning to transfer (from web-collected to surveillance images) the CNN weights. Yu *et. al.* [2] applied the joint Bayesian network to classify the vehicle features extracted from a CNN model. Sochor *et. al.* [18] used an unpacked (using 3D box) image, detected 3D boxes and vehicle viewpoint information as input to train their CNN model. *The method proposed in this research is different, because: (a) it does not use any 3D model based vehicle image synthesis approach and (b) it follows the detection-free approach which considers the complete/holistic scene image.*

The car type classification method proposed by Huttunen *et. al.* [5] is very similar as they also used the *holistic scene images* like the proposed method. They analyzed two methods based on the CNN (AlexNet [39] architecture) and SVM classifiers. Their results verified that the CNN based classifier provides better results than the SVM based method.

Different *image or vision based* vehicle classifiers have been proposed for ATC based on the SIFT descriptors and the estimation of the vehicle occupied area [7], [22], [40]. We believe that this is the first reported research that applies the CNN based and ensemble classification approach for ATC. Moreover, the novel contributions are: (a) introduces a new and challenging *vehicle classification* scenario; (b) proposes a novel vehicle type classifier using an ensemble classification approach; (c) presents a compressed and efficient VGG-16 based CNN model; and (d) clarifies an underexplored (*holistic scene based*) classification approach with CNN.

III. Problem formulation and Vehicle Database

This paper considers the ATC problem within an existing pay tolls setup installed throughout the motorways owned by the *VINCI Autoroutes* company. This problem requires classifying the vehicles into five distinct classes based on certain specifications and physical measurements, such as height, weights and number of axles. The amount of toll payment is determined based on the *detected* class[4] type.

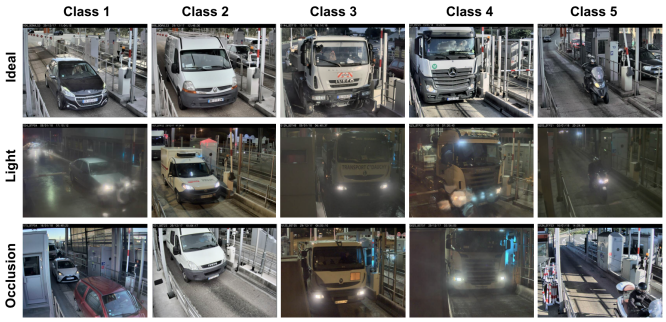

Fig. 3: Illustrations of the images from different classes (in the columns) captured in different conditions (in the rows).

Fig. 3 illustrates examples from different classes, which are primarily distinguished as follows:

Class 1: *light vehicles* - height less than 2 meters.
Class 2: *intermediate vehicles* - height between 2 and 3 meters.
Class 3: *heavy vehicles* - height over 3 meters and have 2 axles.
Class 4: *heavy vehicles* - height over 3 meters and have more than 2 axles.
Class 5: *motorbikes, side-cars and trikes.*

The pay tolls are equipped with several OS, cameras and inductive loops, see Fig. 4 for an illustration. Existing system uses the decisions from these OS to classify the vehicle type. The camera is used to capture image/video, which is later used by the human operators to verify the misclassifications[5]. The inductive loops are used for vehicle detection on the toll in order to trigger the photo capture.

The OS is an apparatus disposed at the entrance (called ***pre-OS***) and the exit (called ***post-OS***) of the toll collection lanes and used to measure the height and the number of axles of the vehicles. Usually it provides the class label as the values: *1,2,3,4 and 5*. However, occasionally it provides *0* or *9* to indicate a missing or inconsistent detection.

The vehicle dataset is collected from the cameras located at the pay tolls. It consists of total 73,638 images: 44,437 of class 1, 8073 of class 2, 11,466 of class 3, 3,262 of class 4 and 6,400 of class 5. Therefore, the samples for different classes are distributed non-uniformly, where class 1 has much larger number of samples compared to others. Fig. 3 illustrates several examples of the images from the dataset.

[4]More details of the class types specification are available at https://www.vinci-autoroutes.com/fr/classes-vehicules.

[5]The clients pay an incorrect amount due to misclassification. In such cases, the operators receive demands for reimbursement. Then, the human observers are engaged to manually verify the vehicle type.

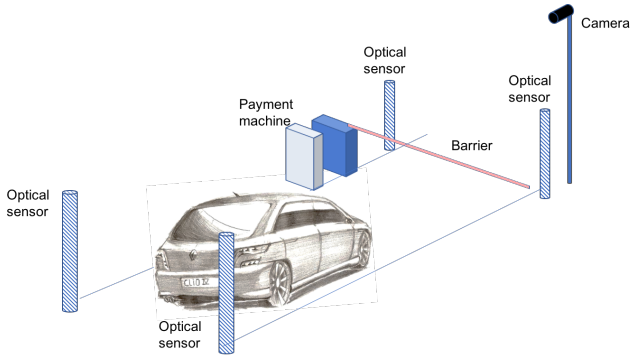

Fig. 4: Existing ATC set-up, where the OS is used to classify vehicles. The camera captures images for later useage.

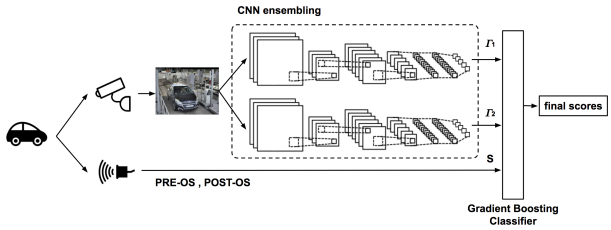

Fig. 5: Proposed vehicle classification method.

IV. Methodology

Fig. 5 provides an illustration of the proposed vehicle classification method. It exploits the existing setup (see Fig. 4) and take the data from both OS and the camera. While the OS directly provides the class label decision, the camera provides the color image. It adopts the CNN [9], [10] based classification strategy to determine the vehicle type from the input color images. Next, it fuses two different categories of classifiers output: (a) the continuous class probabilities from the CNN classifier and (b) the discrete class labels obtained from two (pre and post) OS. The fusion is accomplished using the Gradient Boosting [25] classifier to obtain the vehicle class.

1) ***Convolutional Neural Network****:* The basic architectural ideas of a CNN [9], [10] consist of the *Convolution* and *Pooling* operations. The Convolution operation has the following form:

$$f_{x,y,k}^{C,l} = \mathbf{w}_k^{l\,T} f_{x,y}^{Op,l-1} + b_k^l \tag{1}$$

where $\mathbf{w}_k^l$ and b_k^l are the weights and bias of the k^{th} feature map, $f^{Op,l-1}$ and $f_{x,y,k}^{C,l}$ are the input and output feature maps, l denotes the layer and (x, y) is the spatial image coordinate. C denotes convolution and Op represents various operations, *e.g.*, input (when $l = 1$), convolution, pooling, activation, etc. Pooling applies local operations, *e.g.*, computing average within a local neighborhood has the following form:

$$f_{x,y,k}^{P_{avg},l} = \frac{1}{m \times n} \sum_{(m,n)\in\mathcal{N}_{x,y}} f_{m,n,k}^{Op,l-1} \tag{2}$$

where, $\mathcal{N}_{x,y}$ is the local spatial neighborhood. Often a spatial resolution reduction is applied after pooling. In order to ensure non-linearity, the output from one layer are passed through

	Input	Conv	Max Pool	Conv	Max Pool	Conv	Max Pool	Conv	Max Pool	Conv	Max Pool	Avg Pool	Dropout	FC	Dropout	CL	
Filt Support	224 x 224	3	2	3	2	3	2	3	2	3	2	7	0.5	1024	0.25	SoftMax	Output
Stride		1	2	1	2	1	2	1	2	1	2	1					
Pad		1	0	1	0	1	0	1	0	1	0	0					
Num. Fits		64		128		256		512		512							
Num. Rep.		2		2		3		3		3							

Fig. 6: Illustration of the CNN architecture. *Num. Rep.* indicates the number of consecutive repetition of the same block. *Filt. Support* indicates the size of the convolution kernel.

different activation functions [39], [41], e.g., the Rectified Linear Unit (ReLU): $f_{x,y,k}^{ReLU,l} = max(f_{x,y,k}^{Op,l-1}, 0)$. Besides, several strategies are commonly applied, such as dropout [42] and normalization [43]. A layer with full connections, called Fully Connected (FC) layer, often appears at the end of the concatenated layers. Finally, a loss layer is added in order to optimize the model parameters with respect to a loss function. The widely used Softmax loss has the following form:

$$\mathcal{L}_{Softmax} = -\sum_{i=1}^{N} log\left(\gamma_{iy_i}\right), \text{with, } \gamma_{iy_i} = \frac{e^{\mathbf{w}_{y_i}^T f_i + b_{y_i}}}{\sum_{j=1}^{K} e^{\mathbf{w}_j^T f_i + b_j}} \tag{3}$$

where γ_{iy_i} is the probability of i^{th} sample for being assigned to its true class label y_i. N and K denote the number of training samples and the number of classes. In Eq. (3), the true class label (y_i) can be replaced with arbitrary class label to compute the set of probabilities for all classes.

The proposed method exploits two different CNN models based on the VGG [44] (*CNN-1*) and Inception [45] (*CNN-2*) architectures. *CNN-1* modifies the VGG-16 [44] by: (a) adding average (7×7) pooling after the last max pooling layer; (b) eliminating the last two FC layers; (c) reducing number of neurons in the first FC layer from 4096 to 1024 and (d) adding dropout [42] layers before (0.5%) and after (0.25%) the FC layer. It reduces the complexity[6] of the VGG-16 [44] model ≈ 9 times and comprises only 15M parameters (compared to 134M), see Fig. 6 for the details. *CNN-2* is the Inception-V3 [45] model. Sect. V-A1 provides the details of the CNNs training strategy. These models take an image of size[7] 224 × 224 (CNN-1) and 299 × 299 (CNN-2) as input and provide a 5 dimensional vector of class probabilities as output.

2) ***Gradient Boosting****:* GB [25] is a popular heterogeneous data classification method. It constructs a single strong predictor by iteratively combining the weaker predictors. This combination is achieved by a greedy procedure, where the gradient descent is applied in the function space.

[6]The ATC task of this research needs to classify only 5 classes, which is significantly lower than 1000 classes for which the VGG models were designed, see Table-1 of [44]. This indicate that we do not need such high number of parameters after the last max pooling layer of the VGG model. Besides, it is well known that a CNN model with significantly high number of parameters tends to overfit the training data by memorizing them and hence reduce performance on the test data [10]. Therefore, we design the VGG-14 model with reduced number of parameters so that it converges faster, reduces overfitting problem and improves generalization. In order to validate these empirically, we performed a comparative trial and provide results in Table VI, which show that VGG-14 performs better than VGG-16.

[7]In order to respect the design of a CNN model it is necessary to resize the arbitrary sized image to the model-specific input size. Although it causes the loss of important information (*e.g.,* aspect ratio) the CNN models are yet able to efficiently accomplish the task due to their robustness to scale changes.

Let $(\Gamma_i^t)^{t=1,2} = (\gamma_{ij}^t)_{j=1,2,3,4,5}^{t=1,2}$ be the continuous class probabilities obtained from the CNN-t models, $S_i = (s_i^{pre}, s_i^{post})$ be the discrete decision labels obtained from the pre-OS and post-OS and y_i denote the true class label. Now, let $\mathbf{x}_i = (\Gamma_i^1, \Gamma_i^2, S_i) = (\gamma_{i1}^1, \dots \gamma_{i5}^1, \gamma_{i1}^2, \dots \gamma_{i5}^2, s_i^{post}, s_i^{post})$ be the concatenated feature vector obtained by combining Γ_i^t and S_i. Therefore, $\mathbf{x}_i$ represents the outcome of the ensemble (here by concatanation) applied at the *1st layer* of the proposed method. Next, the GB method is used to accomplish the desired ensemble task at the *2nd layer*. The goal of GB method is to find an approximation of a function $F(\mathbf{x})$ which minimizes the multi-class classification loss $\mathcal{L}_{Multiclass}(y_i, F(\mathbf{x_i}))$ as:

$$\mathcal{L}_{Multiclass} = \frac{\sum_{i=1}^{N} \omega_i log\left(\frac{e^{a_{iy_i}}}{\sum_{j=0}^{M-1} e^{a_{ij}}}\right)}{\sum_{i=1}^{N} \omega_i} \tag{4}$$

where ω_i is the weight, a_i is the value of the target function for the i^{th} sample. The proposed method uses the CatBoost algorithm [26] to perform classification with the GB method. It is chosen because of its efficiency to fuse multiple categories of data, particularly the discrete categorical data. Details of the CatBoost training strategy is provided in Sect. V-A2.

V. Experiments, Results and Discussion

This section begins with the details of experimental settings for training. Then, it evaluates the proposed method and discusses several related issues.

A. *Training*

First, the CNN models are trained to obtain the class probabilities for each image. Next, the GB method is trained and subsequently used to get the final class label. The training dataset is selected by randomly splitting the collected dataset into train/validation/test sets with $70\%/15\%/15\%$ proportion respectively. This split provides 51.5K samples for the training-set and 11K samples for both test and validation sets. Distributions of the samples-per-class on all sets have similarity with the distribution of the entire dataset.

1) CNN Training: The images from the training set are used to optimize the CNN models parameters. The CNN models are initialized with the parameters learned for the ImageNet [24] object classification by the same model. The weighted/balanced Softmax loss [23], [46] is applied as the optimization objective to address the *class imbalance distribution*. These weights correspond to the inverse of class volume. The *L2* regularization is applied on the CNN weights. The learning rate is set to 0.001 and the mini-batch size is set to 100. Data augmentation is applied by horizontally flipping the images. The Stochastic Gradient Descent (SGD) [9] method is used for optimization, which is chosen empirically.

2) GB Training: The training data for the GB method is obtained by concatenating the outputs of the CNN Softmax layer and the one-hot encoded values from OS. The CatBoost [26] classifier is used with the depth set to 6, learning rate set to 0.03 and the maximum number of iterations set to 500.

B. *Results and discussion*

This section evaluates the proposed approach on the test set and compare it with the competitive methods. The classification accuracy is used for the comparison and the precision measure is used for an in-depth class-wise analysis. Additionally, the running time is measured to evaluate time complexity.

This research considers a novel vehicle type recognition use case for ATC. Unfortunately, no existing benchmark is available, except the OS, which are components of proposed method. Therefore, in order to perform a competitive evaluation several alternative methods (which could be applied for this task) are considered as follows:

- **Car type Classification by Huttunen *et. al.* [5] (CCH)**: used the AlexNet [39] for detection-free and holistic scene based car classification.
- **Object of interest classification (*OIC*)**: existing VMMR [2], [4], [6], [11]–[14], [16]–[19] methods mostly follow this approach. In order to compare with this type of methods, a competitive method is developed as follows: (a) crop the object of interest from the database images using the RetinaNet[8] [46] object detector, pre-trained on the COCO dataset [49]; (b) apply an empirical rule to choose the vehicle of interest in case of multiple detection and (c) train a VGG-14 model on the cropped images. Sect. V-C3 provides additional details.
- **Single components of the proposed method**: PRE-OS, POST-OS, InceptionV3(CNN-1) and VGG-14 (CNN-2).

TABLE I: Comparison among the competitive methods using the accuracy (in %) and computation time (in milliseconds).

Method	Acc (%)	Time (ms)
CCH	94.77	7
OIC	94.84	387
Fusion of classifiers (ours)	**99.03**	63
VGG-14 (CNN-1)	95.71	30
InceptionV3 (CNN-2)	95.36	44
PRE-OS	0.10	NA
POST-OS	52.77	NA

The test set accuracies of these methods are reported in Table I, which shows that the proposed method provides the best result (99.03%). Moreover, it not only outperforms the existing deployed solution (52.77%) significantly, but also provides reasonably better results than the alternative CNN-based solutions, *i.e.* CCH (94.77%) and OIC (94.84%). Indeed, the large performance gap ($\approx$ 4%) with the stand-alone CNN-based methods reveals the effectiveness of the proposed ensemble approach. Note that the components of the ensemble are chosen empirically by exploring numerous CNN combinations (see Sect. V-C1 for futher details). The individual CNN components of the proposed method performs better than the competitive methods. The comparison with the CCH method justifies the choice of the CNN models, see Table VII for more results and Sect. V-C2 for additional performance analysis of the VGG-14 (CNN-1) model. Moreover, the comparison with the OIC method justifies the choice of the *detection-free* classification approach.

[8]Other object detectors, such as Faster R-CNN [47] and YOLOv2 [48] were explored and RetinaNet [46] is chosen based on its performance.

TABLE II: Comparison of the different classification methods to fuse the outputs from the CNNs and OS.

Method	RF	SVM	MLP	XGBoost	CatBoost
Acc (%)	98.82	98.90	98.95	98.97	**99.03**

Besides accuracy, these methods are compared with the computation time, that is measured on the NVIDIA K80 GPU machine with 12 GB of GPU-memory. The right-most column of Table I indicates that the proposed appoach is executed within a reasonable computation time and hence is well acceptable for the given ATC task. The OIC approach is the most expensive. Comparison of CCH, CNN-1 and CNN-2 shows that the computation time is related to the complexity of the CNN models. Note that, the proposed method runs two CNNs in parallel, which reduces the time ($\approx$ 17ms).

Next, the chosen ensemble method, *i.e.* CatBoost [26] is briefly evaluated by comparing with several commonly used classifiers, such as RF, SVM, MLP and XGBoost [37]. Note that, the ensemble based on scores averaging is not applicable for the heterogenious outputs obtained from the CNNs and OS. The test-set accuracies reported in the Table II show that the CatBoost method provides the best result.

Next, the accuracy and precision for each vehicle category is studied to analyze the in-depth characteristics of the proposed method and its components. Table III presents the per-class accuracy, from which the observations are: (a) the proposed method provides the best accuracy for classes 1, 2, 3 and 5 and (b) it is 1.63% less accurate than the post-OS for class 4. The performance of pre-OS is very poor[9]. The post-OS performs best on class 4, well on classes 2 and 5 and poor on class 3 and 1. Indeed, its performance on class 3 and 4 indicates that it is biased towards the class 4. The accuracies of the CNN models show that CNN-1 (VGG-14) is better for classes 1,2,3 and 5 and CNN-2 (Inception) is better for class 4. Therefore, the classification strengths of each model for different classes justify their integration within the proposed method. Table IV provides the per-class precision measures, which show that the proposed method gives best precision for all classes. Indeed, from the business perspective this measure is the most suitable metric to evaluate the trustworthiness of a method. Therefore, the high precisions indicate that the predictions from the proposed method are highly reliable.

TABLE III: ***Accuracy*** (%) analysis for each vehicle category.

	1	*2*	*3*	*4*	*5*	*Total*
Proposed	**99.92**	**97.11**	**97.62**	94.91	**99.90**	**99.03**
PRE-OS	0.03	0.25	0.29	0.00	0.10	0.10
POST-OS	45.27	86.07	29.21	**96.54**	82.47	52.77
CNN-1	98.21	88.62	90.65	81.47	99.59	95.36
CNN-2	98.92	90.11	92.74	69.45	99.27	95.71

Next, in Table V the confusion matrix of the proposed method is analyzed to gain further insights on the classification performance. The observations are:

[9]This is due of the source (customer-care centers) of the dataset where the majority of the images correspond to low confidence pre-OS data.

TABLE IV: ***Precision*** analysis for each vehicle category.

	1	*2*	*3*	*4*	*5*
Proposed	**99.66**	**98.66**	**97.22**	**95.88**	**100.00**
PRE-OS	0.10	0.05	0.74	0.00	0.08
POST-OS	95.45	17.90	77.27	92.40	95.10
CNN-1	98.97	86.83	90.08	78.59	99.90
CNN-2	98.65	88.43	89.37	87.66	99.69

TABLE V: Confusion matrix computed from the classification results of the proposed method.

True class	Predicted class					*All*	*Recall*
	1	*2*	*3*	*4*	*5*		
1	6661	2	3	0	0	6666	99.92
2	14	1178	20	1	0	1213	97.11
3	8	14	1681	19	0	1722	97.62
4	0	0	25	466	0	491	94.91
5	1	0	0	0	963	964	99.90
All	6684	1194	1729	486	963	11056	
Prec.	99.66	98.66	97.22	95.88	100.00		

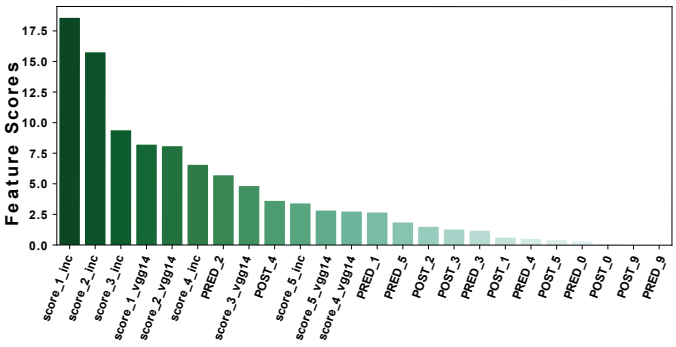

Fig. 7: Feature importance of the CatBoost [26] classifier. Each vertical bar provides the importance score of the corresponding features listed in the horizontal axis.

- For class 1 it produces 0.08% error, which are misclassified as class 2 (0.03%) and class 3 (0.05%).
- For class 2 it produces 2.9% error, which are misclassified as class 1 (1.15%), 3 (1.65%) and class 4 (0.08%).
- For class 3 it produces 2.4% error, which are misclassified as class 1 (0.46%), class 2 (0.81%) and class 4 (1.10%).
- For class 4 it produces 5.1% error, which are misclassified as class 3.
- For class 5 it produces 0.1% error, which are misclassified as class 1.

The results in Table I show that the proposed ensemble approach provides significant (3.32%) improvement compared to its best individual classifier. This encourages to identify the important features to better understand the contribution of the fusion components. Fig. 7 illustrates the CatBoost feature importance calculated based on the training dataset. It shows that the scores from both CNNs significantly contribute to the final decision. Besides, the PRE-OS = 2 and POST-OS = 4 have been identified as the most important OS features. These provides sufficient evidence to realize that the OS inputs are crucial to identify certain vehicle properties (*e.g.* the number of axles) which are difficult to infer from the images.

TABLE VI: Validation accuracy and loss at the end of the training for different CNN architectures.

Method	Acc (%)	Loss
VGG-14	**96.07**	0.146
VGG-16	95.76	0.177
Inception	95.87	0.148
AlexNet	95.25	0.188
Resnet50	95.48	0.156
DenseNet	95.81	0.515
Xception	96.01	**0.132**

There are certain scenarios in which the OS are known to perform relatively poorly for particular classes:

- Vehicles from class 1 may be misclassified as class 2 if it has something on its roof (*e.g.* lights or luggage carrier).
- Vehicles from classes 2 and 5 are often confused and require an operator to correct the misclassifications.
- Vehicles from class 1 with a trailer attached to them should be classified as class 2 by classification rules, but are usually classified as class 1 by the OS.

On the other hand, the image could be inadequate for correct classification. For example, the *same* truck can be classified as class 3 or 4 based on the presence of a trailer behind it, which can be invisible in the image. Likewise, the lack of visibility (due to occlusion or frontal view image only) of the the number of axles in the image causes the OS to become the only reliable source for correct classification. Therefore, the combination of image-based class predictions and the OS labels should outperform the individual components.

C. *Performance analysis*

This section provides more details about several aspects of the proposed method, possible alternatives of the given problem and finally identifies the limitations and future scopes.

1) Selection of the CNN models: In order to select the appropriate components for the proposed ensemble, several CNN models have been explored: VGG-14, VGG-16 [44], Inception [45], AlexNet [39], ResNet50 [50], DenseNet [51] and Xception [52]. These models were trained with the same specifications described in Sect. V-A1. Table VI reports the validation accuracies and loss values of these models. Fusion of the CNN classifiers have been explored with and without including the OS decisions. The results are shown in Table VII, which show that the best combination of the CNN models is *vgg14 + xception* when the OS are *excluded*. On the other hand, the best combination *including* the OS features is *inception + vgg14*. Interestingly, the proposed *VGG-14* model commonly appears in both combinations. Therefore, its characteristics are further investigated with additional experiments.

2) Analysis of the VGG-14 model: Table VIII provides the confusion matrix that represents the details of test set predictions by the VGG-14 model. It provides high (over 90%) recall for all classes except class 4 (69.45%), where the misclassifications are mostly caused by class 3. This can be explained by the visual similarity of the vehicles from these classes, which constitute the trucks and buses, and are distinguished based on the number or axles.

TABLE VII: Test set accuracy for the fusion of models subsets (up to 3 models) and optionally the OS.

Model	Test accuracy	
	without OS	with OS
alexnet	94.77	98.60
densenet161	95.00	98.72
inception	95.36	98.85
resnet50	94.78	98.97
vgg14	95.71	98.95
vgg16	95.42	98.50
xception	95.73	98.82
alexnet + densenet161	95.99	98.79
alexnet + inception	95.89	98.96
alexnet + resnet50	95.58	98.81
alexnet + vgg14	96.08	98.90
alexnet + vgg16	95.65	98.59
alexnet + xception	96.12	98.93
densenet161 + inception	96.15	98.81
densenet161 + resnet50	95.78	98.87
densenet161 + vgg14	96.08	98.82
densenet161 + vgg16	96.01	98.70
densenet161 + xception	96.07	98.78
inception + resnet50	95.72	98.97
inception + vgg14	96.19	**99.03**
inception + vgg16	95.84	98.78
inception + xception	96.00	98.96
resnet50 + vgg14	96.09	99.00
resnet50 + vgg16	95.57	98.61
resnet50 + xception	95.79	98.91
vgg14 + vgg16	95.79	98.60
vgg14 + xception	**96.46**	98.95
vgg16 + xception	95.87	98.68

TABLE VIII: Confusion matrix from the VGG-14 predictions.

True class	Predicted class					*All*	*Recall*
	1	***2***	***3***	***4***	***5***		
1	6594	68	1	0	3	6666	98.82
2	79	1093	41	0	0	1213	90.11
3	3	74	1597	48	0	1722	92.74
4	1	1	148	341	0	491	69.45
5	7	0	0	0	957	964	99.27
All	6684	1236	1787	389	960	11056	95.71

Vision-based visual understanding of the performance can be accomplished with the Gradient-weighted Class Activation Mapping (Grad-CAM) [27] technique. It localizes the attention-map or important regions in the image, which is exploited by the CNN model for classification. Fig. 8 illustrates several examples, where the success/failure of VGG-14 can be explained by the ability to focus on the relevant part of the image for the vehicle-class of interest.

Learning based visual discrimination can be realized with the t-distributed Stochastic Neighbor Embedding (t-SNE) [28] method. A subsample (50%) of the validation set is used to project the 5-dimensional VGG-14 classification scores into the 2-dimensional t-SNE output. Fig. 9 provides the illustration and exhibits the overlaps among several classes: 1-2, 2-3 and 3-4. Indeed, for certain vehicles it is difficult to discriminate the closer classes (*e.g.,* 1-2 and 2-3) when the height measurement acts as the most prominant measure instead of their visual appearance. Likewise, the number of axles is significant to discriminate among the classes 3 and

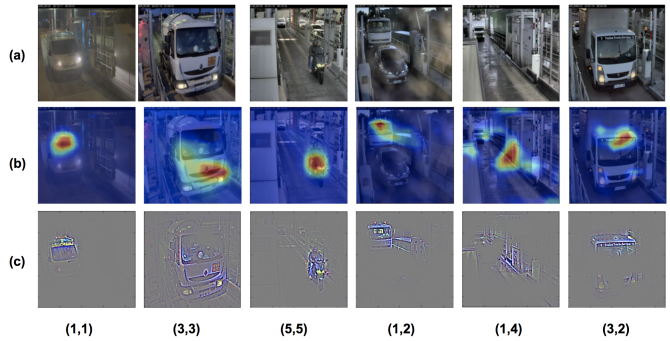

Fig. 8: Illustration of the correctly and incorrectly classified vehicle images. Within each parentheses (t, p), the first value t indicates the *true class label* and the second value p means the *predicted class label*. (a) input image (resized); (b) Grad-CAM visualization and (c) Guided Grad-CAM visualization.

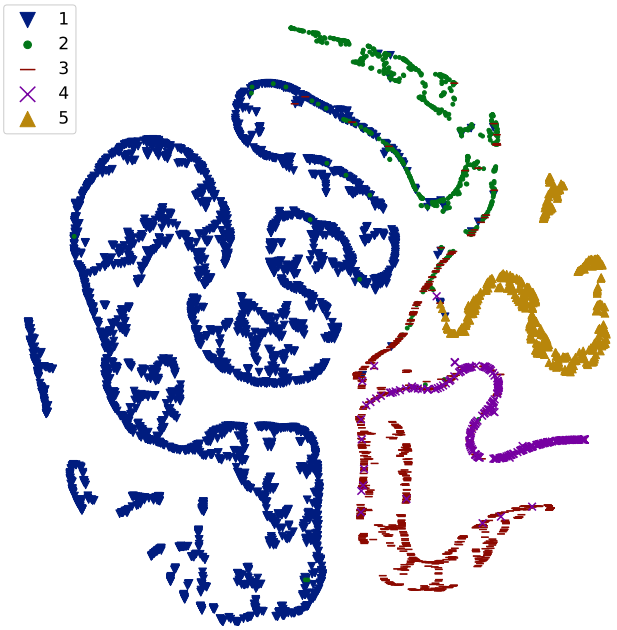

Fig. 9: T-SNE illustration of the VGG-14 output scores for the subsample (50%) of the validation set, where different colors with markers correspond to the true class of the image.

4. In many cases this feature is partially or fully occluded in the image, which leads to misclassification by VGG-14. These image based limitations is difficult to overcome within the existing paytoll setup and hence left the only choice to use additional data from different sources, such as the OS.

3) Object of interest classification: This paper adopts the *holistic scene based object classification* strategy. In this context, the true class is defined by the class of the *vehicle of interest* when multiple objects appear in the image, see Fig. 2 for an example. However, the common approach (followed by the most VMMR methods) is to apply *vehicle detection followed by classification*, which is implemented in Sect. V-B. This sub-section provides additional analysis on this. Note that, a problem is encountered with this approach when the object detector fails, *i.e.,* no object of interest is detected in the images. In such case, two possibilities are explored: (a) *strategy A*: discard the images and (b) *strategy B:* consider the whole image as the vehicle of interest. The VGG-14 model is used to train and classify with the above strategies and the proposed *holistic scene based* classifier is considered as a benchmark for comparison. Table IX provides the test set accuracies and shows the best result ($\approx$ 1% better than the nearest one) is achieved by the proposed *holistic scene based classifier*. For strategy A, two results are obtained: the accuracy only on the nondiscarded images, and the accuracy by considering the discarded images as misclassified. It is observed that, compared to Strategy B the accuracy of strategy A is not better even after discarding the images. This indicates that, in case of the failure of the vehicle detector it is better to consider the entire image to represent the vehicle rather than discarding it. This indeed provide additional evidence to further support the *holistic scene* based classification. Moreover, the significant increase of the computational complexity augmented by the detection method should be taken into account. The combination of both of these facts motivated this research to pursue the *detection-free and holistic scene based classification* approach rather than the *detection followed by classification* approach.

TABLE IX: Accuracy on the test set for scene classification with VGG-14 model and OIC.

Model	**Acc**
Scene classification	**95.71**
OIC, strategy A (only nondiscarded images)	94.75
OIC, strategy A (all images)	90.26
OIC, strategy B	94.84

TABLE X: Matching classes in the simplified problem version, COCO dataset and VINCI Autoroutes classification guidelines.

Simplified classes	**COCO dataset**	**VINCI Autoroutes**
car	car	class 1, 2
bus/truck	bus, truck	class 3, 4
motorcycle	motorcycle	class 5

4) Comparison of the existing solution (OS), scene classification and the out-of-the-box solution: This subsection considers a simplified and immediate solution (from the business point of view), which does not need to collect data and train models. The RetinaNet [46] model is selected for this purpose. However, it was pretrained on the COCO dataset [49] which does not provide the similar class labels required for the ATC task. Therefore, the number of ATC-classes are reduced to three categories: car, bus/truck, and motorcycle, Table X provides further details of the the class labels mapping. Table XI provides the comparison, where the results are obtained for a subset of the test dataset. This subset (constitutes 94% of the dataset) is constructed by considering the images for which the pre-trained RetinaNet [46] model has detected at least one vehicle with the score higher than 50%. This approach with RetinaNet achieves good accuracy for classes *bus/truck* and *motorcycle*. However it performs poorly on the *car* class and provides an overall accuracy of 77.32%, which is less than the POST OS performance. The proposed *holistic scene based*

TABLE XI: Analysis of the accuracy (%) with respect to individual vehicle categories.

Class	PRE-OS	POST-OS	RetinaNet	vgg14
car	75.88	97.38	69.92	**99.45**
bus/truck	19.62	44.06	**99.08**	96.47
motorcycle	0.00	83.10	93.43	**99.30**
All	59.10	86.20	77.32	**98.87**

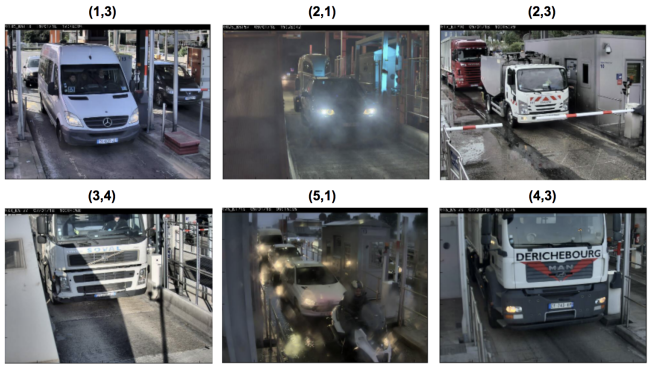

Fig. 10: Illustration of the misclassified vehicle classes. Within each parentheses (t, p), the first value t indicates the *true class label* and the second value p means the *predicted class label*.

classification with VGG-14 model yields the best overall and class-by-class accuracy, except for *bus/truck* class.

5) Limitations of the proposed method: The confusion matrix in Table V provides the misclassified cases. Moreover, manual analysis by human observers is performed to visually inspect the reasons for the misclassifications. Fig. 10 illustrates several examples, from which the main causes are identified as: (a) poor light conditions and occlusion, particularly occlusion of the axles and top of the vehicle; (b) class 2 is often misclassified due to the occluded caravan behind it, which causes the vehicle be categorized as class 2 instead of 1; (c) subtle rules in the class estimation, *e.g.,* symbols on the vehicle that transports people with special needs will be categorized as class 1 instead of 2 or 3. These difficulties constitute additional challenges for the proposed method.

The above analyses indicate several weaknesses of the proposed method. Particularly, it exhibits most of the limitations for class 4 which is misclassified as class 3. In future, these errors can be minimized by following several ways: (a) increase training data by collecting more diverse samples, particularly for classes 3 and 4, and the special rules cases; (b) synthesize more data using data augmentation approaches; (c) incorporate efficient pre-processor to tackle the difficult lighting conditions; (d) enhance the efficiency of the classifier by incorporating discriminative loss functions rather than the ordinary softmax loss; and (e) incorporate deeper CNN models. Besides enhancing the efficiency, the proposed method can be evaluated on a variety of similar vehicle classification tasks from different contexts. Moreover, it will be evaluated on the existing car classification datasets [19], [21] where the objectives are much different compared to the given ATC based classification.

VI. Conclusion

This paper proposes a novel vehicle classification method for a practical use case of ATC, which is currently accomplished with several OS and human operators. The proposal consists of a novel multi-classifier fusion-based method, which combines the classification decisions from the OS and the class probabilities estimated from the camera image using two CNN models. The proposed method significantly outperforms the performance of the existing deployed system by increasing the accuracy from 52.77% to 99.03%. Additionally, it outperforms several alternative *state-of-the-art* CNN based methods, which could be used for the ATC task. Obtained results indicate that the proposed approach can be adapted to a large number of vehicle classification scenario where the classification decision can be made by fusing the outputs from multiple classifiers. The extensive experiments, analysis and discussions provided in this paper indicate several interesting perspectives and challenges for the future work.

References

[1] S.-C. Hsu, I.-C. Chang, and C.-L. Huang, "Vehicle verification between two nonoverlapped views using sparse representation," *Pattern Recognition*, vol. 81, pp. 131–146, 2018.

[2] S. Yu, Y. Wu, W. Li, Z. Song, and W. Zeng, "A model for fine-grained vehicle classification based on deep learning," *Neurocomputing*, 2017.

[3] A. Asvadi, L. Garrote, C. Premebida, P. Peixoto, and U. J. Nunes, "Multimodal vehicle detection: fusing 3d-lidar and color camera data," *Pattern Recognition Letters*, 2017.

[4] M. Biglari, A. Soleimani, and H. Hassanpour, "A cascaded part-based system for fine-grained vehicle classification," *IEEE Trans. on Intelligent Transportation Systems*, 2017.

[5] H. Huttunen, F. S. Yancheshmeh, and K. Chen, "Car type recognition with deep neural networks," in *Intelligent Vehicles Symposium*, pp. 1115–1120, IEEE, 2016.

[6] J. Wang, H. Zheng, Y. Huang, and X. Ding, "Vehicle type recognition in surveillance images from labeled web-nature data using deep transfer learning," *IEEE Trans. on Intelligent Transportation Systems*, 2017.

[7] R. P. Loce, E. A. Bernal, W. Wu, and R. Bala, "Computer vision in roadway transportation systems: a survey," *Journal of Electronic Imaging*, vol. 22, no. 4, p. 041121, 2013.

[8] M. M. Fahmy, "Computer vision application to automatic number plate recognition," *IFAC Proceedings Volumes*, vol. 27, no. 12, pp. 169–173, 1994.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[11] L. Dlagnekov and S. J. Belongie, "Recognizing cars," tech. rep., Dept. of Computer Science and Engineering, University of California, San Diego, 2005.

[12] G. Pearce and N. Pears, "Automatic make and model recognition from frontal images of cars," in *Advanced Video and Signal-Based Surveillance*, pp. 373–378, IEEE, 2011.

[13] J.-W. Hsieh, L.-C. Chen, and D.-Y. Chen, "Symmetrical surf and its applications to vehicle detection and vehicle make and model recognition," *IEEE Trans. onIntelligent Transportation Systems*, vol. 15, no. 1, pp. 6–20, 2014.

[14] H. He, Z. Shao, and J. Tan, "Recognition of car makes and models from a single traffic-camera image," *IEEE Trans. on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3182–3192, 2015.

[15] A. J. Siddiqui, A. Mammeri, and A. Boukerche, "Real-time vehicle make and model recognition based on a bag of surf features," *IEEE Trans. on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3205–3219, 2016.

[16] X. Wen, L. Shao, Y. Xue, and W. Fang, "A rapid learning algorithm for vehicle classification," *Information Sciences*, vol. 295, pp. 395–406, 2015.

[17] J. Fang, Y. Zhou, Y. Yu, and S. Du, "Fine-grained vehicle model recognition using a coarse-to-fine convolutional neural network architecture," *IEEE Trans. on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1782–1792, 2017.

[18] J. Sochor, J. Špaňhel, and A. Herout, "Boxcars: Improving fine-grained recognition of vehicles using 3-d bounding boxes in traffic surveillance," *IEEE Trans. on Intelligent Transportation Systems*, 2018.

[19] L. Yang, P. Luo, C. Change Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," in *Conf. on Computer Vision and Pattern Recognition*, pp. 3973–3981, 2015.

[20] Q. Hu, H. Wang, T. Li, and C. Shen, "Deep CNNs with spatially weighted pooling for fine-grained car recognition," *IEEE Trans. on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 3147–3156, 2017.

[21] H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang, "Deep relative distance learning: Tell the difference between similar vehicles," in *Conf. on Computer Vision and Pattern Recognition*, pp. 2167–2175, 2016.

[22] A. Suryatali and V. Dharmadhikari, "Computer vision based vehicle detection for toll collection system using embedded linux," in *Int. Conf. on Circuit, Power and Computing Technologies*, pp. 1–7, IEEE, 2015.

[23] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognition*, 2017.

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *Int. Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[25] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, pp. 1189–1232, 2001.

[26] D. et. al., "Catboost: gradient boosting with categorical features support," in *NIPS Workshop on ML Systems*, 2017.

[27] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Int. Conf. on Computer Vision*, Oct 2017.

[28] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[29] N. Shvai, A. Meicler, A. Hasnat, E. Machover, P. Maarek, and A. Nakib, "Ensemble classifiers based classification for automatic vehicle type recognition," in *IEEE World Congress on Computational Intelligence*, 2018.

[30] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proc. of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.

[31] M. A. Hasnat, O. Alata, and A. Trémeau, "Joint Color-Spatial-Directional clustering and Region Merging (JCSD-RM) for unsupervised RGB-D image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2255–2268, 2016.

[32] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Information Fusion*, vol. 16, pp. 3–17, 2014.

[33] C. Ju, A. Bibaut, and M. van der Laan, "The relative performance of ensemble methods with deep convolutional neural networks for image classification," *Journal of Applied Statistics*, pp. 1–19, 2018.

[34] G. Heitz, S. Gould, A. Saxena, and D. Koller, "Cascaded classification models: Combining models for holistic scene understanding," in *Advances in Neural Information Processing Systems*, pp. 641–648, 2009.

[35] W. Zhang, Q. Wang, and C. Suo, "A novel vehicle classification using embedded strain gauge sensors," *Sensors*, vol. 8, no. 11, pp. 6952–6971, 2008.

[36] S.-I. Oh and H.-B. Kang, "Object detection and classification by decision-level fusion for intelligent vehicle systems," *Sensors*, vol. 17, no. 1, p. 207, 2017.

[37] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[38] B. Zhang, "Reliable classification of vehicle types based on cascade classifier ensembles," *IEEE Trans. on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 322–332, 2013.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[40] J. Y. Ng and Y. H. Tay, "Image-based vehicle classification system," in *Asia-Pacific ITS Forum and Exhibition*, 2012.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Int. Conf. on computer vision*, pp. 1026–1034, 2015.

[42] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[43] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Int. Conf. on Machine Learning*, pp. 448–456, 2015.

[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. on Learning Representations*, 2015.

[45] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

[46] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Int. Conf. on Computer Vision*, pp. 2999–3007, 2017.

[47] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015.

[48] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Conf on Computer Vision and Pattern Recognition*, pp. 779–788, 2016.

[49] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, pp. 740–755, Springer, 2014.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Conf. on Computer Vision and Pattern Recognition*, 2016.

[51] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer, "Densenet: Implementing efficient convnet descriptor pyramids," *arXiv preprint arXiv:1404.1869*, 2014.

[52] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Conf. on Computer Vision and Pattern Recognition*, pp. 1800–1807, 2017.

Nadiya Shvai received MSc degree in Statistics, in 2010, and PhD in Mathematics, in 2015, from Taras Shevchenko National University of Kyiv, Ukraine. She worked on natural language processing, optimization, and matrix theory. Her main research interests are machine learning, deep learning and linear algebra.

Abul Hasnat received MSc degree from Erasmus Mundus CIMET in 2011 and PhD in "image, vision and signal" from Jean Monnet University, France, in 2014. He worked on RGB-D image segmentation, spectral image reconstruction and document image analysis. His research activities are focused in the area of computer vision, image processing, machine learning and data mining.

Antoine Meicler received a MSc in Machine Learning and Statistics from ISAE - Supaero in 2014. He then applied his knowledge to practical business situations and tackled complex data challenges within startups, consulting firms and big data companies. Since 2016, he is working as a freelance research scientist and is strongly interested in artificial intelligence, machine learning and computer vision.

Amir Nakib received a M.D. in electronics and image processing in 2004 from the University of Paris 6. In 2007, he received a Ph.D. degree in computer sciences from the University of Paris 12. Since 2010, he is associate professor at the university Pars Est Creteil, France. His main research interests are stochastic global optimization and their applications.