Deep Learning Approach to Detection of Preceding Vehicle in Advanced Driver Assistance

Paweł Forczmański(✉) and Adam Nowosielski

Faculty of Computer Science and Information Technology,
West Pomeranian University of Technology, Szczecin, 52 Zolnierska St.,
71-210 Szczecin, Poland
{pforczmanski,anowosielski}@wi.zut.edu.pl

Abstract. In paper we propose a detection method for objects in video stream taken in front of a car by means of deep learning. The successful detection of preceding cars is a part of the analysis of current road situation including emergency and sudden braking, unintentional lane change, traffic jam, accident, etc. We include the results of preliminary experiments employing video stream captured by camera installed behind frontal wind screen. The detection and classification are performed using Convolutional Neural Network preceded by road lane detection. We performed several experiments on real-world data in order to check the accuracy of the proposed algorithm.

Keywords: Deep learning · Convolutional neural network · Vehicle detection · Vehicle classification · Road lane detection

1 Introduction

1.1 Motivation

Nowadays, fast evolution of intelligent transport systems is observable. One of the most interesting areas of such progress is the development of autonomous and self-driving cars. Autonomous car looks like the vehicles used today, having a steering wheel and several seats that face the driving direction. The driver is necessary to supervise such a car, which takes control only in certain situations. Currently, some elements of autonomy are already offered to the customers, i.e. adaptive cruise control, self-parking, and automatic braking [1]. The next, or some say, parallel stage of development is a self-driving vehicle, which potentially takes control over the driving in case of heavy traffic or on highways. Such evolution may lead to the total disappearance of the steering wheel making vehicle drive using the same set of sensors as autonomous vehicles. In such case, the car has to know its exact position, identify objects nearby, and constantly calculate safe and optimal route. This real-time-based situational and contextual responsiveness requires a powerful visual computing algorithms that combine data from all available sensors, while also planning the safest path [2].

Both concepts, however, are founded on the same idea of recognizing environmental conditions and taking decisions about driving. In case of autonomous car the decision

J. Mikulski (Ed.): TST 2016, CCIS 640, pp. 293–304, 2016.
DOI: 10.1007/978-3-319-49646-7_25

is only a suggestion for a driver, who is eventually responsible for the car, while self-driving car does not assume any human actions. Hence, both car types should profit from artificial intelligence and machine vision methods.

The most common problem related to both autonomous and self-driving cars addressed in the literature is the recognition of traffic signs while driving [3]. Fortunately, such an issue is well described and solved in many cases. There are successful implementations in many middle-class cars also. The other problem, which we focus on, is the detection of cars on the road. While this is crucial in terms of driving safety, contemporary car crash-avoidance systems and experimental self-driving have also such systems implemented. However, they often rely on radar and other sensors to detect cars and various obstacles on the road. Hence, the expected improvement is a vehicle/obstacle detection system that can perform in close to real-time based on visual cues only. Such vision-based detection could make systems for recognizing vehicles both cheaper and more effective.

In our algorithm we employ recently proposed deep learning approach. It is associated with a specific branch of machine learning that solves the task of mapping input data such as image features, e.g. brightness of pixels, to certain outputs, e.g. abstract class of object. The deep learning model, in a form of a hierarchical representation of the perceptual problem, compose functions into sets of input transformations, through transitional representations, to output. Such composition constitutes a deep, multi-layered model encoding low-level, elementary features into high-level abstract concepts. The strength of such an approach is that it is highly learnable. The idea of learning comes from a classical neural networks, i.e. we feed an input of a deep network iteratively, and let it compute layer-by-layer in order to generate output, which is later compared with the correct answer. The output error moves backward through the net by back-propagation in order to correct the weights of each node and reduce the error. This operation improves the model during computations. The scheme of processing is presented in Fig. 1.

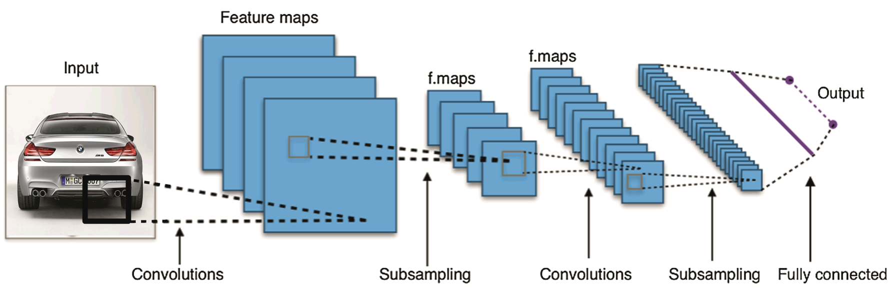

Fig. 1. Data flow in convolutional neural network [own study]

In this paper we propose to use convolutional neural network to solve a problem of multi-class vehicle recognition based on video analysis. While the idea is not purely original, we provide a complete processing flow, including road lane detection, region of interest extraction and final vehicle classification. We propose several improvements in order to make it possible to execute our algorithm close to real-time.

The paper is organized as follows. First we present related works, especially in the field of vehicle detection and classification, from science and industry. Then, we introduce the most important elements of out algorithm including lane detection, region-of-interest definition and vehicle extraction/classification. Finally, we present some exemplary results of numerical experiments and we finish it with some conclusions and future plans.

1.2 Existing Solutions

The general task of detecting certain objects in static scenes or video streams implicates the determination of the image portion containing the investigated object. The selection of appropriate characteristic features used to build an object model is crucial in this case. Additional aspects to be considered are the mechanism of feature matching and the method of scanning input scene. In most cases, the process of detecting specific objects does not rely on an information about the detected object, i.e. object pose, size and position. Further difficulties may result from the scene illumination changes, environmental conditions and specific look of objects [4].

The solution of the problem of detecting vehicles in images can be based on various assumptions related to scene and observer characteristics. If the scene is static and the camera observes moving vehicles, the solution can benefit from background modelling methods [5, 6]. In such an approach, moving objects are segmented from background and then classified [7]. That improves the performance, since the detection can be the most time-consuming task. On the other hand, when the camera is mounted on a moving vehicle, the background modelling cannot be applied, since the scene changes constantly and the extraction of moving objects is complex. Hence, the algorithm have to scan whole scene area with a sliding window and compare extracted regions with a template [5]. In case when the initial information about interesting object is absent, it is required to perform searching in all possible locations and using all probable window (or image) scales. It may significantly increase the computational complexity. Therefore, it is profitable to limit the scanning area for a classifier and the size of sliding window basing on some heuristics. It can employ the knowledge about road area, horizon level and the distance of observation.

The scientific literature contains many proposals of visual features used as a determinant of vehicle presence. Most of them use Local Binary Patterns, Histogram of Oriented Gradients, Spectral Features from Discrete Cosine Transform etc. In [8] an exemplary method for vehicle detection is proposed. It models an unknown distribution of vehicles by means of higher order statistics (HOS). Finally, the classification is performed using HOS-based decision measure. Another interesting algorithm is presented in [9]. It project all input pixels' colors into a new feature space. Bayesian classifier is applied to classify pixels belonging to the vehicles. Finally, Harris corner detector is applied to find extreme corners, and mark a vehicle. The authors of [10] presented a detection method that extracts rear-views vehicles. It is important, that it does depend on the road boundary or lane position. The segmentation of the region of interest employs shadow area under the vehicles and simple low-level features such as edges and symmetries. Vehicle detection combines statistics-based and knowledge-based methods. The number of false detections is reduced using the a'priori knowledge about detected vehicles. Finally, Support Vector Machine is used for two-class classification problem. A similar application of an ensemble of several one-class SVMs is presented in [11].

Many recent solutions for the vehicle detection, e.g. [12, 13], employ extractors based on Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and Haar-like features, passed to the certain cascading classification algorithm, e.g. AdaBoost. The HOG represents direction of brightness changes in some area, while LBP labels pixels by thresholding, resulting in a binary sequence. In case of simple rectangular Haar-like features, the difference of the sum of pixels in areas inside the rectangle is calculated. All of them are rather easy to calculate yielding fast computations and real-time operation.

The alternative direction in research is associated with a use of deep learning methods to create robust classifiers without a need of carefully crafted low-level features. An interesting study is presented in [14]. The authors collected a large volume of road data and applied computer vision and deep learning algorithms to solve problems related to car and lane detection. They showed that convolutional neural networks (CNNs) may be used to perform such tasks in real-time. As it was shown, deep learning can be applied in autonomous driving. Another exemplary approach based on deep learning is presented in [15]. The authors propose to extract the frontal view of a car to recognize the maker and the model. After that, they transform the frontal view of a car to its feature mapping using principal component analysis (PCA). Finally, they use deep learning with three layers of restricted Boltzmann machines (RBMs). Another novel approach is presented in [16], where a deep learning based vehicle detection algorithm with two-dimensional deep belief network (2D-DBN) is proposed. The authors developed a 2D-DBN architecture that employs second-order planes as input and bilinear projection yielding enhanced accuracy in vehicle detection. According to the authors, the experimental results showed that their method works better than provided state-of-the-art algorithms.

Besides scientific-only approaches, there are also several commercial products available on the market. They are based on various principles, however most of them works in a similar way. The earliest solutions to preceding vehicle detection were algorithms implemented in Lexus LS (2006), Volvo s80 (2007), Lincoln MKS/MKT (2009) and Audi A8 (2010). Nowadays, more other carmakers introduce similar systems, both relying on radar and CCD cameras. An independent company - Mobileye [17] - proposed to use an information about detected lane and an additional image information to give a trustworthy threat assessment and collision anticipation. It provides an early detection based on optic flow analysis. Unfortunately, the details are not publicly available and the performance of the developed system is not easy to evaluate. Nevertheless, one of the most advanced systems is the development platform for autonomous cars offered by Nvidia Corp. – Drive PX [18]. It provides driver assistance technologies powered by deep learning, fusion of sensor data, and surround vision. According to the developers, Drive PX can work with video streams captured by 12 cameras, together with lidar/radar, and ultrasonic data. Such information is processed by the appropriate algorithms (based on Deep Neural Networks) in order to precisely analyze the whole environment around the vehicle, including both dynamic and static objects.

As it can be seen, most of presented systems are rather complex and very specialized solutions, dependent on many additional sensors, closely integrated with the car subsystem, hence not to be applied in general purpose, low-end devices, like smartphones or tablets. Therefore in the paper we present an algorithm, that uses only a simple monocular camera and a dedicated software, making proposed solution widely applicable. Thanks to software-only architecture presented system can be easily extended or modified, mainly in the field of vehicle classification.

2 System Overview

2.1 Algorithm Outline

Developed system works on a video stream, assuming a camera is placed behind the windscreen, looking forward. The view area covers a road in front of the host vehicle, with a horizon line roughly in the middle of a picture. The scheme of processing is presented in Fig. 2. It consists of two main modules: road lane detection and object classification. The first module is intended to extract a region of interest (road lane occupied by the host vehicle), while the second one detects vehicles within this region. The algorithm works in a loop, where each iteration ends with a detection/classification.

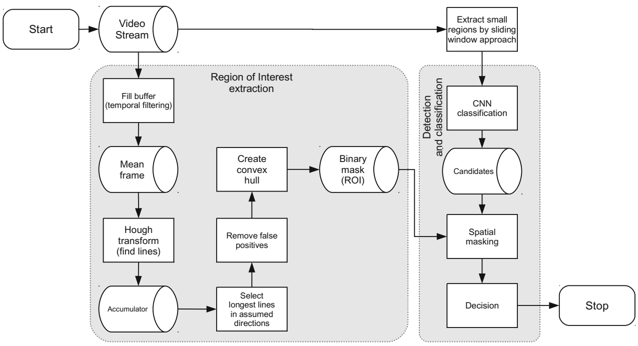

Fig. 2. Scheme of processing [own study]

2.2 Road Lane Detection

The first stage of processing involves the detection of road lines and, thus, the area (Region of Interest), in which we identify vehicles [19]. We assume, that the road in front of the host vehicle is constantly observed. For our purposes, a mobile device equipped with a simple camera (e.g. smartphone or tablet) is appropriate, hence it is very popular and general-purpose. In order to make the calculations easier, we assume that the horizon line is detected with help of a typical accelerometer and g-sensor, however there are many algorithms that are based on visual features only [20]. Our lines detection is based on a well-known Hough transform, however there are other possible algorithms to employ [21].

In order to detect road lines constituting a road lane, we propose to use the following set of operations:

1. Calculate an average of 10 successive frames in order to remove image artefacts and noise;
2. Convert the result into grayscale;
3. Detect edges using Canny detector and create binary image;
4. Remove values above calculated horizon level;
5. Detect lines using Hough Transform (HT) and leave only detected lines that extend at an angle range of -65 to 65 degrees;
6. Select long lines by locating the highest peaks in the HT matrix;
7. Fill gaps in detected lines;
8. Create convex hull based on the endpoints of the line segments;
9. Create binary mask – road lane in front of a vehicle.

Further, obtained region of interest is grown, in order to cover all possible locations of vehicles in front of the host vehicle. Several exemplary detections are presented below (Fig. 3). As it can be seen, the method is resistant to the illumination changes, camera position and horizon level. However, if the road changes its direction in a rapid manner (e.g. sharp curve occurs) the area cannot fully approximate the area in front of the car, since it assumes rather long line segments.

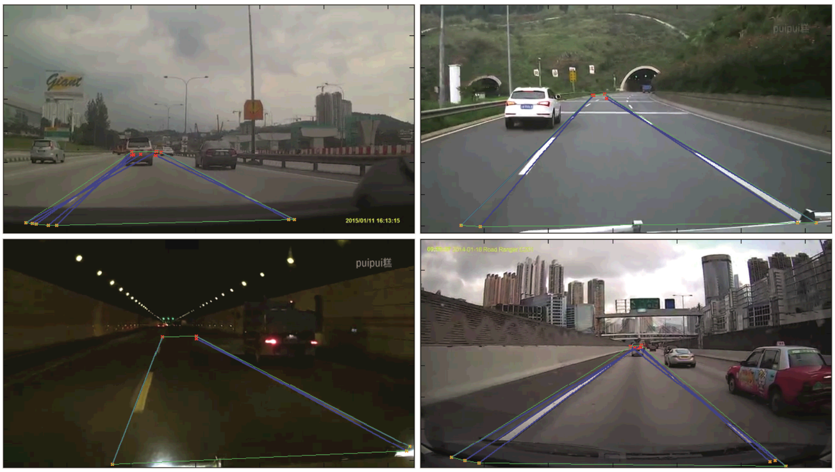

Fig. 3. Examples of road lanes detection in different road conditions [own study]

2.3 Deep Learning Based Vehicle Classification

At the stage of detection/classification we use Convolutional Neural Network, originally known as the NeoCognitron [22] and then advanced by LeCun et al. as LeNet [23]. An algorithm applied in this case is based on a successful AlexNet [24].

The key computational problem in a CNN is the convolution of a feature detector with an input signal. In such an approach, the features flow from single pixels to elementary primitives like vertical and horizontal lines, circles, and color areas. In contrary to the traditional filters that work on single-channel images, CNN filters process all the input channels. Since convolutional filters are translation-invariant, they produce a strong response wherever a specific feature is discovered. In our approach we employed pre-trained network (based on ImageNet) and the MatConvNet toolbox [25] for implementing convolutional neural networks. The CNN used in our algorithm consists of 37 layers. The input layer (which actually contains input image) is 224 × 224 elements. The structure is presented in the Tab. 1, where ‘type’ represents the type of a layer (*cnv* - convolution; computes the output of neurons that are connected to local regions, *relu* - Rectified-Linear and Leaky-ReLU; an elementwise activation function; *mpool* – Max Pooling, performs downsampling operation along the spatial dimensions; *sftm* – performs SoftMax regression) (Table 1).

Table 1. The structure of CNN used for detecting vehicles [own study]

layer	1	2	3	4	5	6	7	8	9	10	11	12	13
type	**cnv**	**relu**	**cnv**	**relu**	**mpool**	**cnv**	**relu**	**cnv**	**relu**	**mpool**	**cnv**	**relu**	**cnv**
support	3x3	1x1	3x3	1x1	2x2	3x3	1x1	3x3	1x1	2x2	3x3	1x1	3x3
stride	1	1	1	1	2	1	1	1	1	2	1	1	1
padding	1	0	1	0	0	1	0	1	0	0	1	0	1
out dim	64	64	64	64	64	128	128	128	128	128	256	256	256
filt dim	3	n/a	64	n/a	n/a	64	n/a	128	n/a	n/a	128	n/a	256
rec. field	3	3	5	5	6	10	10	14	14	16	24	24	32

layer	14	15	16	17	18	19	20	21	22	23	24	25	26
type	**relu**	**cnv**	**relu**	**mpool**	**cnv**	**relu**	**cnv**	**relu**	**cnv**	**relu**	**mpool**	**cnv**	**relu**
support	1x1	3x3	1x1	2x2	3x3	1x1	3x3	1x1	3x3	1x1	2x2	3x3	1x1
stride	1	1	1	2	1	1	1	1	1	1	2	1	1
padding	0	1	0	0	1	0	1	0	1	0	0	1	0
out dim	256	256	256	256	512	512	512	512	512	512	512	512	512
filt dim	n/a	256	n/a	n/a	256	n/a	512	n/a	512	n/a	n/a	512	n/a
rec. field	32	40	40	44	60	60	76	76	92	92	100	132	132

layer	27	28	29	30	31	32	33	34	35	36	37
type	**cnv**	**relu**	**cnv**	**relu**	**mpool**	**cnv**	**relu**	**cnv**	**relu**	**cnv**	**sftm**
support	3x3	1x1	3x3	1x1	2x2	7x7	1x1	1x1	1x1	1x1	1x1
stride	1	1	1	1	2	1	1	1	1	1	1
padding	1	0	1	0	0	0	0	0	0	0	0
out dim	512	512	512	512	512	4096	4096	4096	4096	1000	1000
filt dim	512	n/a	512	n/a	n/a	512	n/a	4096	n/a	4096	n/a
rec. field	164	164	196	196	212	404	404	404	404	404	404

Employed, pre-trained network can distinguish between 1000 general classes of various images (objects). In our case we focused on a subset of 30 objects that can be found on roads, namely (the number in the brackets is the original class number): ambulance (408); amphibious vehicle (409); beach wagon (437); tandem bicycle (445); taxi (469); convertible (512); fire truck (556); freight car (566); garbage truck (570); go-kart (574); jeep (610); limousine (628); minibus (655); minivan (657); motor scooter (671); mountain bike (672); moving van (676); passenger car, coach, carriage (706); pickup truck (718); police van, police wagon (735); racing car (752); recreational vehicle (758); school bus (780); sports car (818); streetcar, tram (830); tank, armoured combat vehicle (848); tow truck (865); tractor (867); trailer truck (868); trolleybus (875).

The scheme of processing the final results from CNN classifier is presented in Fig. 4. In our algorithm we scan every video frame with a sliding window and present each extracted sub-image to the CNN classifier. We take into consideration 30 particular outputs from the network – each one representing probability of a relevant class (listed above). Each single answer falls within a range (0,1>, while the sum of all answers in each location cannot exceed one. Hence, the next stage is voting, which creates accumulated results map that is further superimposed over a region of interest obtained at the previous stage of processing. The voting works as follows. If the sum of all answers is higher than 0.5, then we assume a vehicle in the analysed location. Next, we look for the highest activation and check, if it is larger than 0.1, then we can decide about predicted vehicle class. The binary mask from lane detection stage serves as a filter, which tells vehicles that are on the same road lane from vehicles that are on the neighbouring lanes. Moreover, object that are detected over a horizon line are rejected as a false positives.

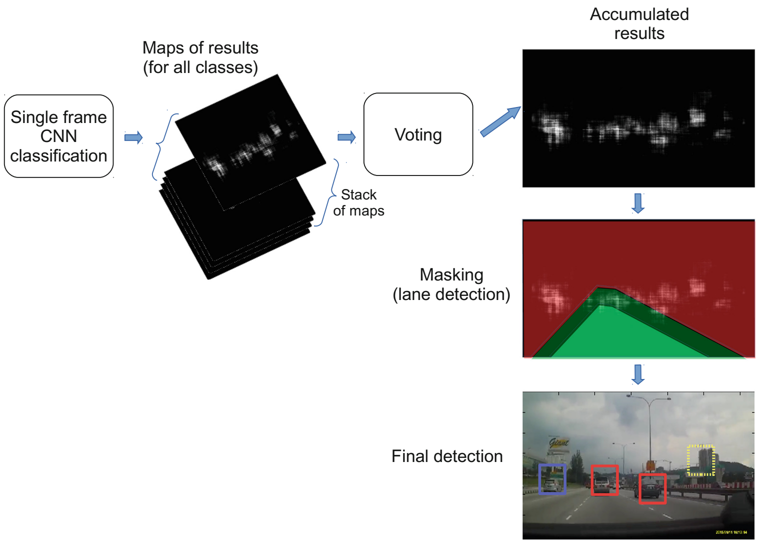

Fig. 4. Final detection stages: classification, voting and masking [own study]

3 Experiments

We performed experiments on data collected from driving recorders working in various lighting conditions and surroundings. The frame-rate was equal to 30 frames per second, with a spatial resolution of a single frame equal to 640 × 360 pixels, 24-bit RGB. A simulation was performed using a MATLAB prototype.

Below in Figs. 5 and 6 one can see some exemplary results of our algorithm. The road lane is marked yellow and the detected/classified vehicles are marked red or blue, depending on the lane position. The vehicle being on the same lane as the host car is marked red, while the other cars are marked blue. In each case, the class number is provided above each marking. The probability level above each extracted vehicle is also shown.

Fig. 5. Exemplary detections in good lighting conditions [own study]

In most cases vehicles are detected and the class is predicted correctly, however, when detected car is far from the camera the classification can be ineffective. It is caused by not ideal representation of samples in the learning set, which, as it should be stressed out, was created for general purpose object classification only.

As it can be seen, poor lighting conditions do not influence the detection of road lane and the detection of vehicles, however, the classification accuracy in this case is degraded (e.g. a bus was misclassified as a trailer truck, a van was recognized as a sports

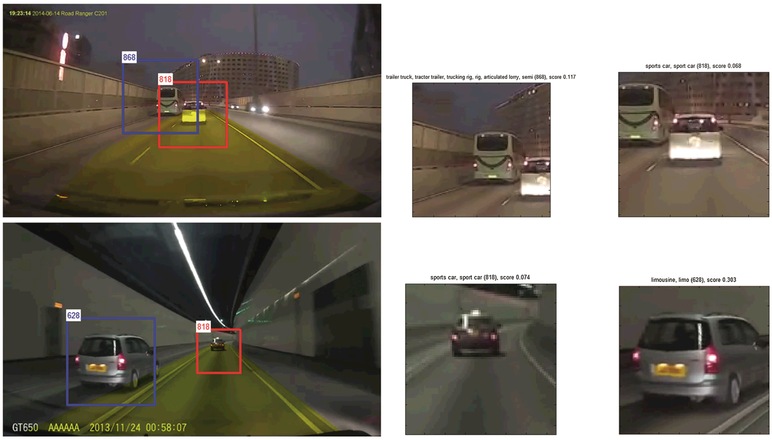

Fig. 6. Exemplary detections in poor lighting conditions [own study]

car and a taxi cab was recognized as a sport car, too). Nevertheless, the detection accuracy stands at a very high level, which makes proposed solution ready for a practical implementation.

4 Conclusion

In the paper, the problem of vehicle detection in real driving conditions was analysed. The purpose of the performed experiments was to verify the possibility of employing deep learning approach for the processing on visual-only data that come from monocular camera. Proposed algorithm does not rely on any dedicated hardware and can be successfully implemented in general-purpose simple computer system, e.g. mobile device. The experiments were performed on video sequences captured during driving in different outdoor conditions. As it was shown, the vehicle detection based on the proposed assumptions is possible and quite successful. The classification fails in case of large distance from the camera and not-perfect representation of a learning set. The future works would include increasing the quality of learning set as well as more precise lane detection stage (especially in case of road bends).

As it was mentioned, many prototypes, like Google's experimental self-driving cars currently rely on a wide array of radar/lidar, and other sensors to detect vehicles and other objects on the road [2]. Elimination of some of that sophisticated equipment could make such cars cheaper and easier to design. On the other hand, presented system may be implemented in low-end mobile devices e.g. smartphones and tablets. Moreover, not just driverless cars that would benefit; modern crash-avoidance systems found in existing cars could also potentially make use of such an algorithm.

References

1. Berger, C.: Continuous integration for cyber-physical systems on the example of self-driving miniature cars. In: Bosch, J. (ed.) Continuous Software Engineering, pp. 117–126. Springer, Heidelberg (2014)
2. Google Inc: Google self-driving car project. https://www.google.com/selfdrivingcar. Accessed 10 Feb 2016
3. Forczmanski, P., Malecki, K.: Selected aspects of traffic signs recognition: visual versus RFID approach. In: Mikulski, J. (ed.) Activities of Transport Systems Telematics. CCIS, vol. 395, pp. 268–274. Springer, Heidelberg (2013)
4. Mazurek, P., Okarma, K.: Application of Bayesian a priori distributions for vehicles' video tracking systems. In: Mikulski, J. (ed.) Transport Systems Telematics. CCIS, vol. 104, pp. 347–355. Springer, Heidelberg (2010)
5. Frejlichowski, D., Gościewska, K., Nowosielski, A., Forczmański, P., Hofman, R.: Detecting parked vehicles in static images using simple spectral features in the 'SM4Public' system. In: Kamel, M., Campilho, A. (eds.) ICIAR 2015. LNCS, vol. 9164, pp. 489–498. Springer, Heidelberg (2015). doi:10.1007/978-3-319-20801-5_54
6. Forczmański, P., Seweryn, M.: Surveillance video stream analysis using adaptive background model and object recognition. In: Bolc, L., Tadeusiewicz, R., Chmielewski, L.J., Wojciechowski, K. (eds.) ICCVG 2010. LNCS, vol. 6374, pp. 114–121. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15910-7_13
7. Okarma, K., Mazurek, P.: Application of shape analysis techniques for the classification of vehicles. In: Mikulski, J. (ed.) Transport Systems Telematics. CCIS, vol. 104, pp. 218–255. Springer, Heidelberg (2010)
8. Aarthi, R., Padmavathi, S., Amudha, J.: Vehicle detection in static images using color and corner map. In: International Conference on Recent Trends in Information, Telecommunication and Computing ITC 2010, pp. 244–246 (2010)
9. Rajagopalan, A.N., Burlina, P., Chellappa, R.: Higher order statistical learning for vehicle detection in images. In: Proceedings of the 7th IEEE International Conference on Computer Vision, vol. 2, pp. 1204–1209 (1999)
10. Wen, X., et al.: A rear-vehicle detection system for static images based on monocular vision. In: 9th International Conference on Control, Automation, Robotics and Vision, pp. 1–4 (2006)
11. Cyganek, B., Woźniak, M.: Pixel-based object detection and tracking with ensemble of support vector machines and extended structural tensor. In: Nguyen, N.T., Hoang, K., Jędrzejowicz, P. (eds.) ICCCI 2012. LNCS (LNAI), vol. 7653, pp. 104–113. Springer, Heidelberg (2012). doi:10.1007/978-3-642-34630-9_11
12. Sun, D., Watada, J.: Detecting pedestrians and vehicles in traffic scene based on boosted HOG features and SVM. In: Proceedings of the IEEE 9th International Symposium on Intelligent Signal Processing (WISP), pp. 1–4 (2015)
13. Tang, Y., et al.: Vehicle detection and recognition for intelligent traffic surveillance system. Multimedia Tools Application (online) (2015)
14. Huval, B., et al.: An Empirical Evaluation of Deep Learning on Highway Driving, arXiv: 1504.01716v3 (2015)
15. Gao, Y., Lee, H.J.: Deep learning of principal component for car model recognition. In: Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV), pp. 48–51 (2015)
16. Wang, H., Cai, Y. Chen, L.: A vehicle detection algorithm based on deep belief network. The Scientific World Journal, vol. 2014, Article ID 647380 (2014). http://dx.doi.org/10.1155/2014/647380. Accessed 1 Dec 2015

17. Mobileye, N.V.: Headway Monitoring and Warning. http://www.mobileye.com/technology/applications/vehicle-detection/headway-monitoring-and-warning/. Accessed 10 Feb 2016
18. Nvidia Inc.: Drive PX, Accelerating the Race to Self-driving Cars. http://www.nvidia.com/object/drive-px.html. Accessed 10 Feb 2016
19. Chang, H.Y., Chih-Ming, F., Huang, C.L.: Real-time vision-based preceding vehicle tracking and recognition. In: 2005 Proceedings of IEEE Intelligent Vehicles Symposium, pp. 514–519 (2005)
20. Kumar, A.M., Simon, P.: Review of lane detection and tracking algorithms in advanced driver assistance system. Int. J. Comput. Sci. Inf. Technol. (IJCSIT) **7**(4), 65–78 (2015)
21. Lech, P., Okarma, K., Fastowicz, J.: Fast machine vision line detection for mobile robot navigation in dark environments. In: 8th International Conference on Image Processing and Communications (IP and C), Advances in Intelligent Systems and Computing, vol. 389, pp. 151–158 (2016)
22. Fukushima, K.: Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. **36**(4), 193–202 (1980)
23. Lecun, Y., et al.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
24. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. **25**, 1106–1114 (2012)
25. Vedaldi, A., Lenc, K.: MatConvNet: CNNs for MATLAB. http://www.vlfeat.org/matconvnet/ (2014)