Neural Network-Based Vehicle and Pedestrian Detection for Video Analysis System

Pavel V. Babayan, Maksim D. Ershov, Denis Y. Erokhin
Department of Automation and Information Technologies in Control
Ryazan State Radio Engineering University (RSREU)
Ryazan, Russia
aitu@rsreu.ru

Abstract—**In our research we compare various neural network architectures that are used for object detection and recognition. In this work vehicles and pedestrians are considered objects of interest. Modern artificial neural networks are able to detect and localize objects of known classes. This allows them to be used in various technical vision systems and video analysis systems. In this paper we compare three architectures (YOLO, Faster R-CNN, SSD) by the following criteria: processing speed, mAP, precision and recall.**

Keywords-video analysis, image processing, object detection, pattern recognition, neural networks, machine learning

I. INTRODUCTION

Smart video processing systems are now widely used in various areas of human life [1, 2]. The development of such systems is associated with enhancement of the computer technology and with the development of new methods for video processing and analyzing. The main tasks that underlie most applications of video processing systems are the detection, recognition and tracking of objects. The solution of these problems lies in the basis of such applications as systems for automatic detection and tracking of objects, robotics, security systems, and video analysis systems.

This paper is dedicated the problem of using artificial convolutional neural networks in the problem of detecting and localizing objects of defined classes.

Before the advent of specialized neural networks for object detection was used approach based on HOG features and a linear SVM [3].

Nowadays for object detection and classification usually used convolutional neural networks. This is due to several reasons:

- Significant progress in the development of graphics processors.
- Large amount of training data.
- Better results in comparison with classical approaches.
- A large number of specialized software packages for data preparation, training and the use of neural networks.

The work has been supported by scholarship of the President of the Russian Federation for young scientists and postgraduate students (SP-2578.2018.5)

978-1-7281-1740-9/19/$31.00 ©2019 IEEE

Neural network architectures for object detection and recognition can be divided into two large groups:

1. Region-based convolutional neural network (R-CNN).
2. Architectures that process the entire image (YOLO, SSD).

II. YOLO (YOU ONLY LOOK ONCE)

The works [4, 5, 6] present the architecture of a neural network object detector called YOLO. The YOLO architecture was originally designed for real-time object detection. In the YOLO algorithm the image is divided into cells using a grid. The probability of the presence of an object is estimated for each grid cell.

In the YOLO algorithm the detection results are presented as a $7 \times 7 \times 1024$ tensor. The estimated probability of the presence of an object of each class in the current bounding rectangle is the product of an estimated probability of presence of an object in a cell and an estimated probability for a particular class.

YOLOv3 detects objects on three scales which made it possible to increase the quality of detection of small objects. YOLOv3 can also assign multiple labels to detected objects. For example, the output labels may be “Pedestrian” and “Child” which are not mutually exclusive and the sum of the outputs may be greater than 1. In YOLOv3 the softmax activation function [7] is replaced with independent logistic classifiers to calculate the probability of an output belonging to a specific label.

III. FASTER R-CNN (FASTER REGION-BASED CONVOLUTION NEURAL NETWORK)

Faster R-CNN [8] is currently one of the frequently used architectures based on deep learning for solving the object detection problem. R-CNN [9] and Fast R-CNN [10] are the forerunners of this architecture.

Processing by R-CNN consists of three main stages:

1. The input image is divided into regions in which objects can be located. The selective search algorithm [11] is used for this purpose. This algorithm

generates 2000 different areas that contain objects with highest probability.

2. Each region is fed to the input of a corresponding trained convolutional neural network which extracts a feature vector for its region.
3. Feature vectors are fed to a set of SVMs that perform the classification function. Each SVM is trained to determine one class of objects. In addition linear regression is used to refine the parameters of the bounding box.

An additional step can be considered the non-maximum suppression algorithm for eliminating the excess number of boxes covering the same object.

The R-CNN architecture achieved high object detection accuracy but disadvantages such as high memory, training and processing time costs were noted. Therefore architecture modifications have been proposed that led to the development of Fast R-CNN:

- The entire image is fed to the input of one convolutional neural network that performs feature extraction. The selection of region proposals is carried out on the basis of a feature map.
- The set of SVMs performing the classification function has been replaced by a softmax layer.

Thus, the convolutional neural network is used once for the entire image instead of processing 2000 intersecting regions. It is also sufficient to train one network with a softmax layer without additional training of many SVMs.

The Fast R-CNN architecture has a significant speed advantage over R-CNN. But another drawback was the algorithm for selecting region proposals (selective search). A modification of this stage led to the creation of Faster R-CNN.

The selective search algorithm has been replaced by region proposal network (RPN). The input of this network is the area of size $n \times n$ taken from the feature map. Then the result is fed to two fully connected layers: box-regression and box-classification. Region proposals obtained using the RPN are represented by the bounding box coordinates and the probability of an object presence in a given region. The probability is calculated using the softmax function.

The Faster R-CNN architecture currently provides high object detection accuracy and is considered relatively fast. At the same time, the main idea of the original R-CNN architecture is preserved: selection of regions where the objects are located and classifying the content of these regions.

IV. SSD (SINGLE SHOT MULTIBOX DETECTOR)

The SSD architecture [12] provides a significant increase in processing speed compared to the Faster R-CNN. If the Faster R-CNN performs the selection of region proposals and the classification in two separate stages, the SSD performs these actions simultaneously when processing the entire image. The SSD operation can be described as follows:

1. The input image passes through a series of convolutional layers which results in a set of feature maps for different scales (for example, 19×19, 10×10, 5×5, etc.).
2. A 3×3 convolution filter is applied at each point of each feature map to produce multiple boxes.
3. The spatial shift and the probability of an object presence are estimated for each box simultaneously.
4. True bounding boxes are compared with predicted ones to eliminate false detections at the learning stage.

Unlike the R-CNN in which there is a minimal probability of an object presence in a region proposal, the SSD does not have filtering step. As a result, a much larger number of boxes are generated at different scales compared to R-CNN, and most of them do not contain an object. In order to solve this problem in the SSD, firstly, non-maximum suppression is used to merge similar boxes into one. Second, the hard negative mining technique is used [13]. According to this technique, only a fraction of the negative examples are used at training iteration. In the SSD the ratio of negatives to positives is 3 to 1.

The selection of region proposals and the classification are performed simultaneously: for a given number of classes C each box is associated with a $(4+C)$-dimensional vector, which contains 4 coordinates and probabilities for all classes. The softmax function is used at the last stage to classify objects.

V. EXPERIMENTAL RESULTS

Five neural network-based object detectors were trained for the purpose of comparison:

1. YOLOv3.
2. Faster R-CNN with InceptionResnet-2 network for feature extraction.
3. Faster R-CNN with Resnet-101 network for feature extraction.
4. SSD with MobileNet-1 network for feature extraction.
5. SSD with MobileNet-2 network for feature extraction.

About 6 700 images with marked up objects of the “pedestrian” and “car” classes were used for training. 750 images were processed during the experiment.

To assess the quality of detector depending on the training iteration the Average Precision (AP) metric is calculated for each class of objects. AP – the average maximum precision for different values of recall. Graphs for different detectors are presented in Fig. 1. The graphs also present the mAP (mean Average Precision) metric, which is the mean AP value for all classes of objects.

Fig. 2 presents precision-recall curve for different detectors. The threshold in the object detection algorithm ranges from 0 to 1 for plotting. The threshold is understood as the minimum value of estimated probability at which the decision on the object detection will be made.

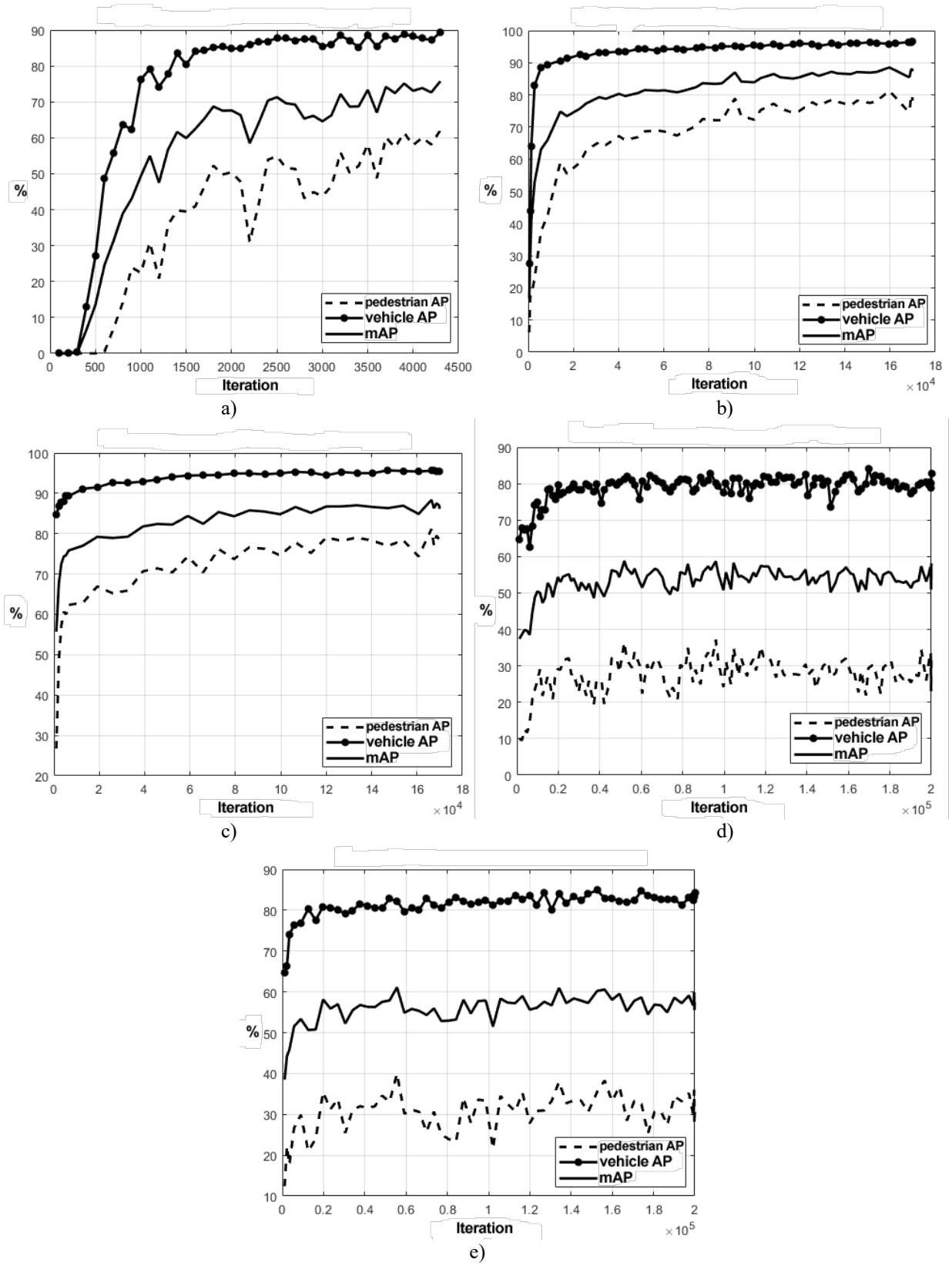

Figure 1. AP and mAP at various iteration:
a –YOLOv3; b – Faster R-CNN+InceptionResnet-2; c – Faster R-CNN+Resnet-101; d – SSD+MobileNet-1; e – SSD+MobileNet-2

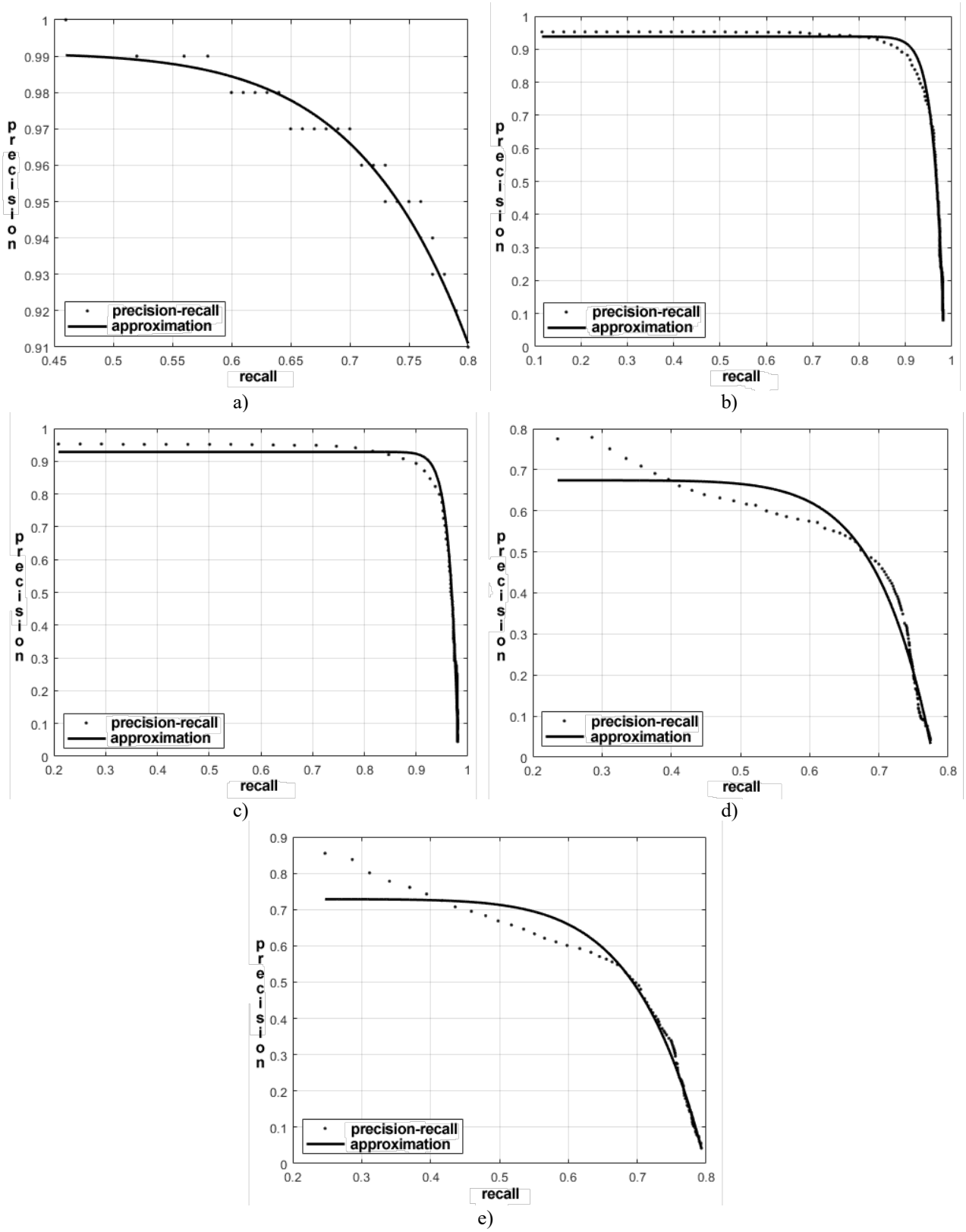

Figure 2. Precision-recall curve:
a –YOLOv3; b – Faster R-CNN+InceptionResnet-2; c – Faster R-CNN+Resnet-101; d – SSD+MobileNet-1; e – SSD+MobileNet-2

The area under the precision-recall curve (AUC) and mAP was used as integral assessments of the detector quality. Also, an evaluation of the computational efficiency of each detector was carried out. For this purpose a personal computer with the NVIDIA GeForce GTX 1070 graphics processor was used and the average processing time for frame with a resolution of 720 × 468 was measured. Named criteria are shown in Table I.

TABLE I. RESULTS FOR DESCRIBED DETECTORS

Detector	Criteria		
	AUC	*mAP, %*	*Time, ms*
SSD + MobileNet-1	0.534	57.204	56
SSD + MobileNet-2	0.573	61.249	58
YOLOv3	0.882	75.740	76
Faster R-CNN + Resnet-101	0.695	86.411	89
Faster R-CNN + InceptionResnet-2	0.722	88.376	119

It should be noted that the processing time is highly dependent on the specific hardware configuration. Also, the processing time using the graphic processors of personal computers does not always reflect the operating time on the mobile device. For example, MobileNet-2 on mobile devices is faster than MobileNet-1, but on the personal computer a slight advantage was obtained for the first version of this neural network.

Fig. 3 presents examples of processed images with detected objects of interest (vehicles and pedestrians). KITTI [14] and Cityscapes [15] datasets were used in our experiments.

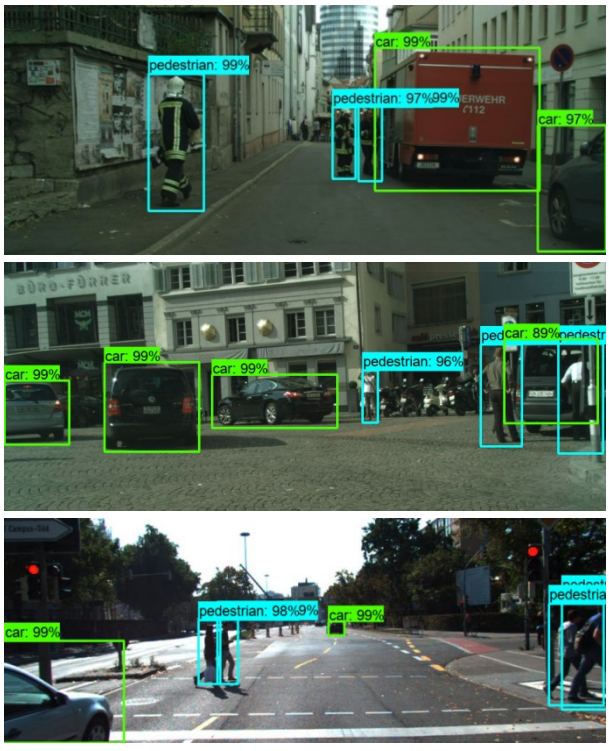

Figure 3. Examples of vehicle and pedestrian detection

VI. CONCLUSION

In this paper we consider three neural network architectures: R-CNN (processing of regions in an image), YOLO and SSD (processing of an entire image). Experimental research of quality and computational efficiency performed using neural network-based detectors YOLOv3, Faster R-CNN with InceptionResnet-2, Faster R-CNN with Resnet-101, SSD with MobileNet-1, and SSD with MobileNet-2. During the experiment we used images containing objects of the classes "pedestrian" and "car".

Faster R-CNN networks have demonstrated an advantage in accuracy. So, according to the results of the experiment, Faster R-CNN based on the InceptionResnet-2 network has the highest accuracy but the average image processing time for this detector is much longer. The SSD architecture is most suitable for image processing in real time (especially when using MobileNet networks) but it must be borne in mind that high accuracy requirements cannot always be met. The neural network-based detector YOLOv3 has a mean accuracy and computational efficiency compared with other detectors.

REFERENCES

[1] Lukyanitsa A.A., Shishkin A.G. Digital video processing. – Moscow: Ai-Es-Es Press, 2009. – 518 p.

[2] B.A. Alpatov, P.V. Babayan, "Image processing and recognition technologies in on-board technical vision systems," Vestnik of Ryazan State Radio Engineering University, No. 2, 2017, pp. 34–44.

[3] B.E. Boser, I.M. Guyon, V.N. Vapnik, "A training algorithm for optimal margin classifiers," Proceedings of the fifth annual workshop on Computational learning theory, ACM, 1992, pp. 144–152.

[4] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You only look once: Unified, real-time object detection," Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[5] J. Redmon, A. Farhadi, "YOLO9000: better, faster, stronger," arXiv preprint, arXiv:1612.08242, 2016, 9 p.

[6] J. Redmon, A. Farhadi, "YOLOv3: An incremental improvement," Tech report, arXiv:1804.02767, 2018, 6 p.

[7] C.M. Bishop, Pattern Recognition and Machine Learning. New York, Springer-Verlag, 2006, 738 p.

[8] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Extended tech report, arXiv:1506.01497, 2016, 14 p.

[9] R.B. Girshick, J. Donahue, T. Darrell, J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2014, 21 p.

[10] R. Girshick, "Fast R-CNN," IEEE International Conference on Computer Vision (ICCV), 2015, 9 p.

[11] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, A.W.M. Smeulders, "Selective Search for Object Recognition," International Journal of Computer Vision, Vol. 104, 2013, pp. 154–171.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, Ch.-Y. Fu, A.C. Berg, "SSD: Single Shot MultiBox Detector," European Conf. on Computer Vision (ECCV), Vol. 9905, Springer, Cham, 2016, pp. 21–37.

[13] S. Wan, Z. Chen, T. Zhang, B. Zhang, K. Wong, "Bootstrapping Face Detection with Hard Negative Examples," arXiv:1608.02236, 2016, 7 p.

[14] A. Geiger, P. Lenz, R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," Conference on Computer Vision and Pattern Recognition (CVPR), 2012, 8 p.

[15] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," CVPR, 2016, 11 p.