International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6, August, 2019

Applicability of Computer Vision Architectures and Their Influence on Traffic Safety of Autonomous Vehicles

Denis Vladimirovich Endachev, Pavel Alexandrovich Vasin, Sergey Sergeyevich Shadrin

Abstract*: This article considers modern rapid architectures of detecting neural networks, structural peculiarities of each selected neural network architectures are analyzed. Experiment is carried out on the basis of potentially dangerous situation upon autonomous vehicle movement; in the selected experimental environment a set of architectures for computer vision system of autonomous vehicle is analyzed, and traffic safety of autonomous vehicle is estimated under various weather conditions; computing time required for application of additional control and analysis algorithms is evaluated. Experimental results are analyzed aiming at development of reasonable selection of neural network architectures for object recognition required for variability of support of autonomous vehicle traffic. Conclusion about applicability of the considered neural network architectures is made for conditions of certain project.

Keywords*: automobile, autonomous wheeled vehicle, unmanned vehicle, neural networks, computer vision, machine vision, traffic safety.

I. INTRODUCTION

Nowadays, as a consequence of development of neural network training, the algorithms of data analysis using neural networks attract attention of researchers in the field of development of autonomous vehicles [1] since modern computing facilities facilitate operation of these algorithms in real time while retaining high flexibility and accuracy.

Conventional methods of object recognition were based on manually obtained properties and pre-neural trained algorithms. Their ultimate performance is achieved by means of arrangement of complicated algorithmic sets which combine analysis of array of low-level image attributes with high-level context from object detectors and camera area classifiers.

Figure 1 illustrates algorithm of computer vision system meeting the requirements described in Specifications of the project 14.624.210049 KamAZ BE.

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

Denis Vladimirovich Endachev*, Federal State Unitary Enterprise Central Scientific Research Automobile and Automotive Institute "NAMI" (FSUE «NAMI»), Moscow, Russia. Email: endachev.d.v@mail.ru

Pavel Alexandrovich Vasin, Federal State Unitary Enterprise Central Scientific Research Automobile and Automotive Institute "NAMI" (FSUE «NAMI»), Moscow, Russia.

Sergey Sergeyevich Shadrin, Federal State Budgetare Education Institution of Higher Education Moscow Automobile and Road Constraction State University (MADI), Moscow, Russia.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

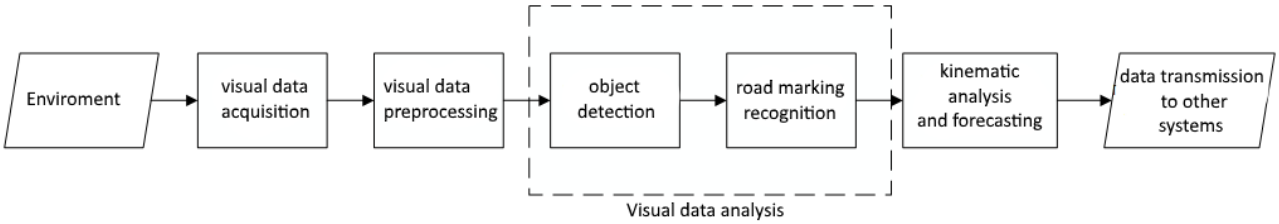

Fig. 1: Algorithm of computer vision system.

In order to provide sufficient information about environment, visual information from sensors of computer vision system should be preprocessed and analyzed several times by means of various algorithms depending on type of the required information. Each step of processing and analysis of visual information increases delay between acquisition of raw information from sensors and acquisition of information ready for application in other systems in order to control autonomous vehicle, to support navigation and localization, etc.

Despite various difficulties of implementation and drawbacks of computer vision systems [2], at present the computer vision algorithms can solve such problems with high rate and accuracy which cannot be achieved by other systems, solution of such problems is obligatory for development of autonomous vehicle.

One of such obligatory problems to be solved for autonomous traffic of vehicles is detection of other traffic users and objects determining vehicle behavior in the road: such as road markers, traffic lights, etc. [3, 4].

Contribution of software module of obstacle detection to delay of data analysis of computer vision is discussed in this work.

II. METHODS

A. Block Diagram: Object detection in computer vision system of autonomous wheeled vehicle

Object detection is the recognition of predefined set of object classes and description of location of each detected object by means of bounding box.

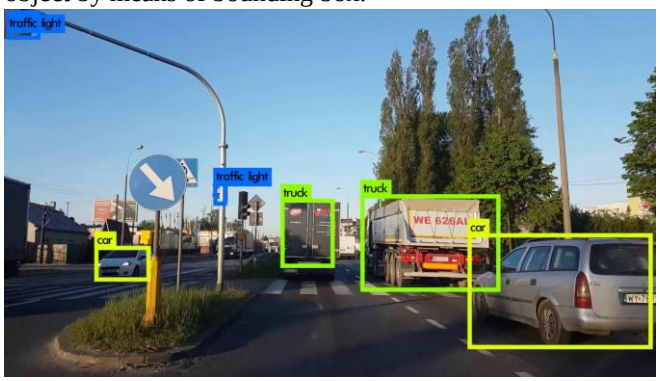

Fig. 2: Operation of object detection algorithm

Rapid development of deep learning made it possible to implement more powerful tools to solve problems existing in conventional architectures and capable to learn semantic, high-level and deeper attributes. These models behave differently depending on network architecture, learning strategy, optimization, etc. At the same time application field imposes certain limitations on properties of system which can be applied in the considered field.

Autonomous vehicles should meet stringent requirements to traffic safety, therefore, the network model used in vision system of unmanned vehicle should provide high accuracy of object detection while maintaining possibility to work in real time together with other vehicle systems [5].

B. Algorithm

Since classical metrics [6] determining operation quality of neural networks do not reflect information about minimum object size required for detector activation, and analysis of architecture is insufficient for exact description of advantages or disadvantages of this or that network in the considered problems, vision systems were experimentally studied based on rapid architectures of neural networks: YOLOv3, Faster R-CNN, Mobilenet, Mobilenet v2, learned by COCO dataset as containing mainly objects at sufficient distance from observer [6], aiming at determination of maximum distance at which they are capable to detect object and on the basis of this information to derive conclusion about their applicability as a component of computer vision system of unmanned vehicle.

YOLO neural network and its modifications

In comparison with other networks of classification by regions (for instance, **Fast RCNN**), which perform detection on the basis of various proposals by regions and, finally, provide multiple forecast for various areas in image, **YOLO** architecture is more similar to **FCNN** (Fully Convolutional Neural Network) and transfers n×n image one time via **FCNN**, receiving m×m forecast at output. This architecture separates input image by m×m grid and generates for each grid cell two bounding boxes and probability of classes for these bounding boxes. It is highly probable that bounding box is larger than a grid cell.

The network has 24 convoluting layers with two succeeding fully connected layers. Instead of inception modules, used by **GoogLeNet**, 1×1 reduction layers are used succeeding by 3×3 convoluting layers. Schematic view is illustrated in Fig. 3.

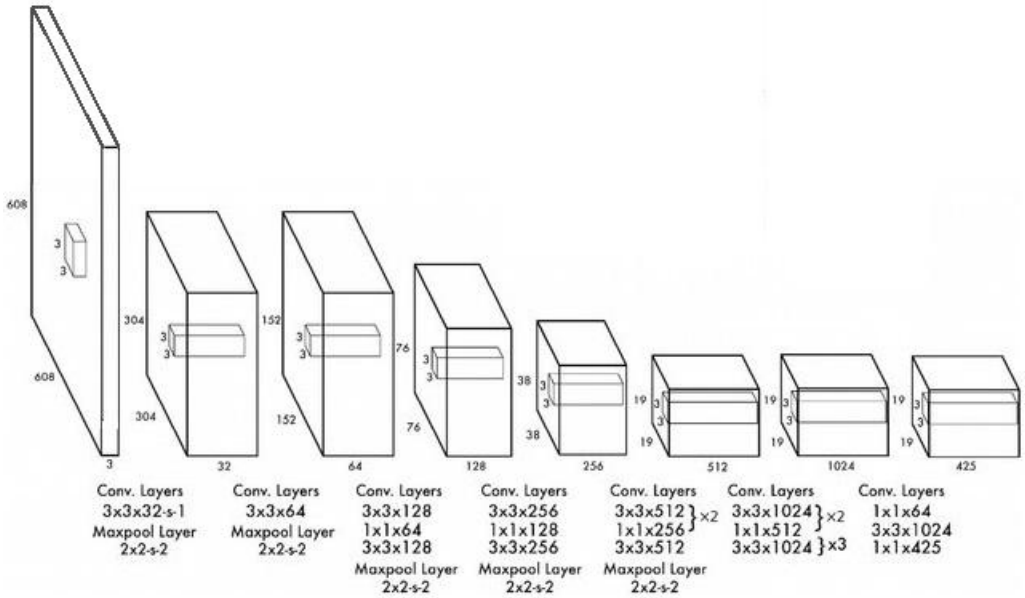

Fig. 3: YOLO architecture.

Fast YOLO uses neural network with lower number of convoluting layers (9 instead of 24) and lower number of filters in these layers. Except for the network size, all training and testing parameters are the same both for **YOLO** and **Fast YOLO**. One convoluting network simultaneously forecasts the set of bounding boxes and probability of classes for these boxes. **YOLO** is trained by complete images and directly optimizes detection productivity. This unified network is characterized by several advantages in comparison with conventional models of object detection.

Firstly, **YOLO** is characterized by high running speed. Since this architecture determines detection as regression, no complicated algorithm of image analysis and conversion is required. In order to detect objects, the neural network is just launched for new image. Basic network variant runs at the speed of 45 frames per second [6] without batch normalization on the basis of Titan X processor, and fast variant **Fast YOLO** operates at the speed of more than 150 frames per second. It means that stream video can be processed in real time with delay less than 25 μs.

Secondly, while forecasting, **YOLO** analyzes overall image instead of its separate segments. Contrary to procedures based on the use of sliding windows and regions, **YOLO** detects overall image during training and testing, hence, it implicitly encodes context data on classes as well as their appearance.

Thirdly, **YOLO** is trained to generalized object representations. While training on the basis of natural images and testing arts objects, **YOLO** is superior to the best detection methods, such as **DPM** and **R-CNN** [7,8]. Since **YOLO** operates closely with generalizations, it is less susceptible to failures upon application to new fields or unexpected input data.

YOLO network uses attributes of overall image for forecasting each bounding box, it forecasts all bounding boxes for all classes in image simultaneously.

YOLO v3 uses Darknet, which initially has 53-layer structure trained by Imagenet. In order to solve detection problem, 53 layers are added to it, which gives 106-layer completely convoluted basic structure for YOLO v3. This explains low speed of YOLO v3 in comparison with YOLO. YOLO v3 operates at the speed of 30 frames per second [6] without batch normalization on the basis of Titan X graphic processor. Schematic view is illustrated in Fig. 4.

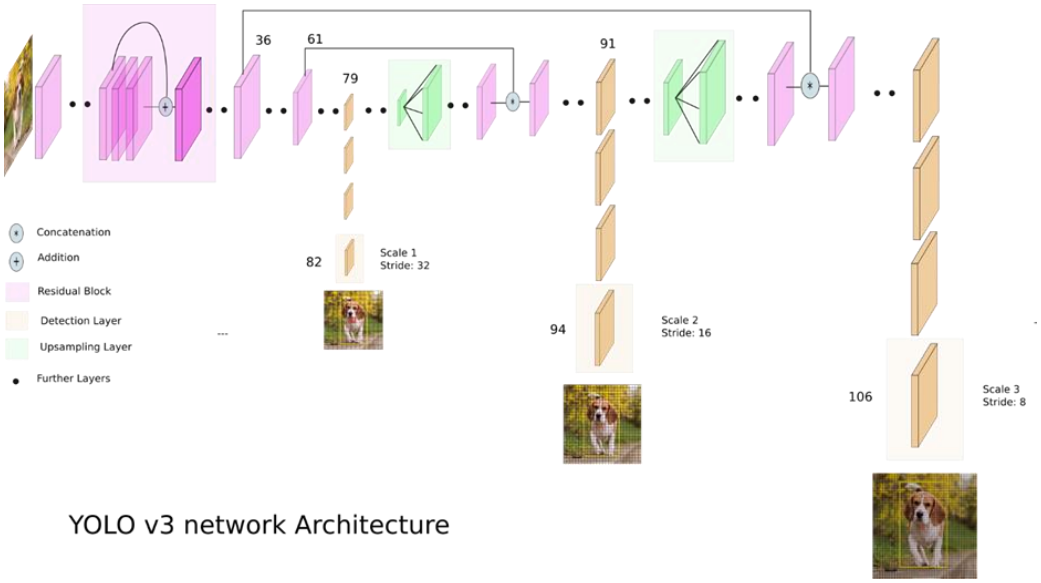

Fig. 4: YOLO v3 architecture

Faster R-CNN

At the conceptual level, Faster-RCNN is comprised of 3 neural networks: Feature Network, Region Proposal Network (RPN), and Detection Network [9]. Schematic view of the network is illustrated in Fig. 5.

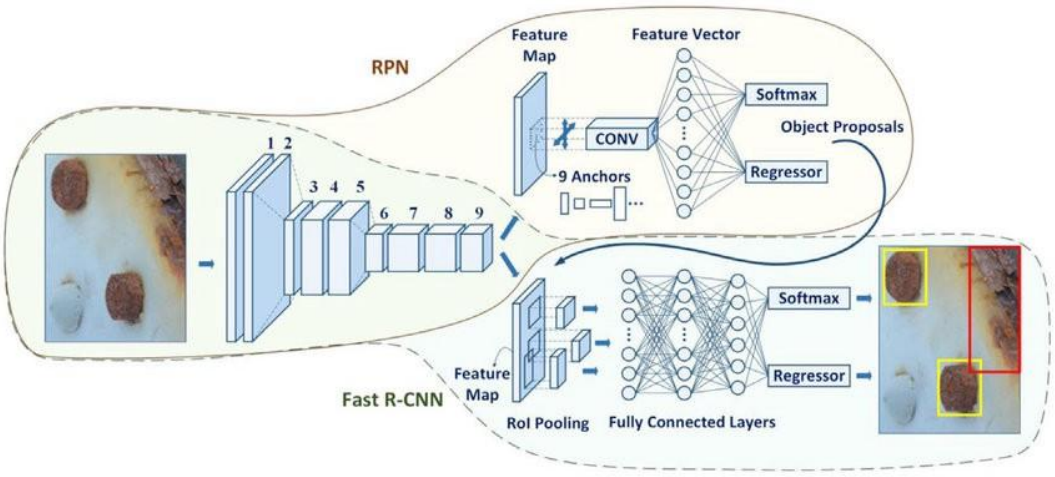

Fig. 5: Faster R-CNN architecture

Feature Network is usually pre-trained network of image classification, for instance, such as VGG, without several last upper layers. The function of this network is generation of Feature maps on the basis of images. Output data of this network retain form and structure of initial image (pixels in initial image are approximately compared with respective pixels in feature maps).

Region Proposal Network (RPN) receives the set of feature maps as input data and generates a set of rectangle proposals by objects, each of them has objectness score: it defines probability that proposal box contains object of each class. In order to classify only internal part of bounding boxes, the feature maps are cropped according to bounding boxes.

Detection Network (also known as RCNN) receives input data from feature network and from RPN, and generates final detection of class and bounding box. Usually it is comprised of four fully-connected layers or dense layers.

SSD -Single Shot Detector

SSD architecture is comprised of convolutional network with direct propagation which generates a set of bounding boxes of fixed size and scores for existence of class objects in this blocks, then follows Non-Maximum Suppression to obtain final detections. Initial layers of the network are based on arbitrary standard architecture used for image classification truncated to classified layers which is referred to as basic network [10]. Then auxiliary structure is added in order to generate detections with key features described below:

Additional convolutional layers are appended to the end of truncated basic network. These layers are consecutively decreased in sizes facilitating detection in several scales. Convolutional model for detection forecast is different for each layer of feature maps contrary to YOLO, which operates in one scale of feature maps.

Each additional block of feature maps can produce fixed set of detection predictions based on a set of convolutional filters.

SSD model appends several layers of feature maps to the end of basic network, which predict displacements by default for bounding boxes of various scales and aspect ratio as well as related confidence score.

Each set of bounding boxes is interrelated with its cell of feature map for certain set of feature map in top part of network. The boxes are located by default so that position of each box with respect to its respective cell in feature map is fixed. In addition, each cell of feature map forecasts displacements and deformation of the boxes with respect to cells in feature map as well as reliability estimations for each class, which determine existence of specimen of each class in each of these boxes. For each block, reliability estimations of class detection are predicted together with four displacements with respect to initial box: vertical displacement, horizontal displacement, variation of width and height of bounding box.

C. Flowchart

Two SSD modifications are considered in this work: with MobileNet v1 and MobileNet v2 as basic network.

MobileNet

MobileNet V1 model is based on depthwise separable convolutions, its flowchart is illustrated in Fig. 6, they are a form of factorized convolutions which convert standard convolution into depthwise convolution and 1×1 convolution known as pointwise convolution.

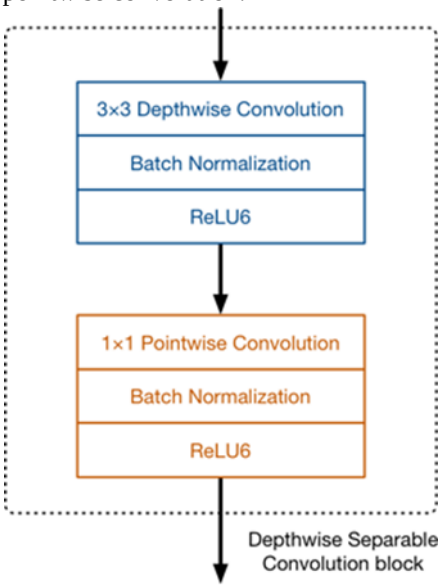

Fig. 6: Separable convolution flowchart

Depthwise convolution applies one filter to each input channel. Then pointwise convolution applies 1×1 convolution for depthwise combination of outputs. Standard convolution both filters and combines inputs to a new set of output in one step. Depthwise separable convolution separates it into two layers: a separate layer for filtration and a separate layer for combination. This factorization results in significant reduction of computations and model sizes. Standard convolution provides filtration of elements based on convolution cores and combination functions so that to generate new data representation [11].

MobileNet V2 also uses depthwise separable convolutions but in this case a block is comprised of three convolution layers. The latter two are the same depthwise convolution which filters input data with succeeding layer of 1×1 pointwise convolution. However, this 1×1 layer performs another function. In V1 the pointwise convolution either retained the same number of channels or doubled it. In V2 it acts reversely: decreases the number of channels. This layer is known as projection layer, it projects data with high amount of measurements (channels) into tensor with lower number of measurements. This layer is also known as bottleneck layer because it decreases the number of measurements passing the network [12]. The bottleneck residual block was named after it, its schematic view is illustrated in Fig. 7.

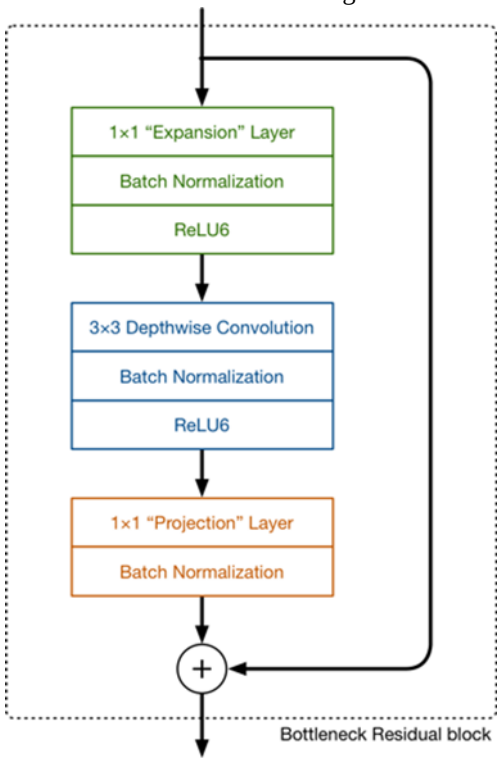

Fig. 7: Bottleneck residual block.

III. RESULTS

A. Experimental results of neural network architectures

Since computer vision system both in terms of hard- and software is high-modular, readily transferable and weakly related with vehicle basic structure, with the aim of simplification, this system was experimentally studied on the basis of Lada Vesta vehicle.

In order to determine distance at which neural network can detect object in camera image, the networks of aforementioned architectures were run on unmanned vehicle computer vision system comprised of a Basle acA780-75gc camera with Basler Lens c125-0418-5m and NVidia Jetson TX2 computer. Table 1 summarizes processing time of one camera frame and maximum distance at which the object was detected: vehicle with confidence score >0.3 for each architecture; test images are exemplified in Fig. 8.

Fig. 8: Example of test images.

Table 1: Architecture testing under actual conditions

Description	Processing time, ms	Distance, m
SSD Mobilenet v1	34	35.8025
SSD Mobilenet v2	33	41.135
Faster R-CNN	131	31.635
YOLOv3	196	46.5425

The processing time and detection distance imply restrictions on movement of autonomous vehicle. The vehicle should move at such speed so that to maintain safety of each road user [13, 16].

Let us consider potentially hazardous situation during movement of autonomous vehicle: stationary vehicle on the path of autonomous vehicle without possibility of avoidance maneuver.

In this case collision could be avoided by emergency braking, if the vehicle speed under current cohesion between tires and surface is lower than that at which stopping distance equals to the distance to object upon its detection.

Vehicle stopping distance is determined by the equation [14], and with consideration for braking features of autonomous vehicle, this equation is rewritten as follows:

$$S_0 = \left(t_0 + t_1 + t_2 + 0.5 \cdot t_3\right) \cdot v_a + \frac{v_a^2}{2g\varphi}, \quad (1)$$

where t_0 is the delay between actual time of event and its fixation by camera; t_1 is the time of visual data processing by neural network; t_2 is the delay upon activation of brakes, t_3 is the time of increase in deceleration (depending on design and technical state of brakes it can vary from 0.05 to 2 s [7]); v_a is the initial vehicle speed; φ is the averaged coefficient of cohesion of pneumatic tires with surface; g is the acceleration of gravity.

Let assume for vehicle with air brake system $t_0 = 0.15$ s, $t_2 = 0.2$ s, $t_3 = 0.5$ s, as well as $\varphi = 0.8$ for dry asphalt, 0.45 for wet asphalt, 0.2 for ice coated asphalt [15]. We neglect aerodynamic properties, surface irregularities, and road inclinations.

Then, the maximum speed at which it is possible to avoid collision by braking in the described situation using each considered architecture is summarized in Table 2.

Table 2: Safe maximum speed

Description	v_{max}, km/h, dry asphalt	v_{max}, km/h, wet asphalt	v_{max}, km/h, ice coated asphalt
SSD Mobilenet v1	72.75	56.77	39.38
SSD Mobilenet v2	78.8	61.35	42.45
Faster R-CNN	65.42	51.6	36.19
YOLOv3	80.74	63.42	44.3

It should be mentioned that in order to increase allowable vehicle speed, it would be required to apply more efficient computer with the same architectures of neural networks for object detection.

According to the project specifications, allowable vehicle deceleration equals to 2 m/s². In this case Eq. (1) of autonomous vehicle stopping distance is rewritten as follows:

$$S_0 = \left(t_0 + t_1 + t_2 + 0.5 \cdot t_3\right) \cdot v_a + \frac{v_a^2}{2a}, \quad (2)$$

where a is the allowable vehicle deceleration. Let us assume further that vehicle deceleration is always 2 m/s² and road coating allows to brake with such deceleration.

Then, the stopping distances for each considered architecture at the speed of 25 km/h are summarized in Table 3.

Table 3: Stopping distance at 25 km/h

Description	Stopping time, m
SSD Mobilenet v1	15.39
SSD Mobilenet v2	15.39
Faster R-CNN	16.07
YOLOv3	16.52

On the basis of previous tables, it is possible to determine free computing time (see Table 4) available for additional analysis of visual data stream and for making decision about necessity to perform emergency braking, thus providing guaranteed traffic safety of autonomous vehicle:

$$t_{free} = \frac{S_{det} - S_{stop}}{v}, \quad (3)$$

where t_{free} is the free computing time, S_{det} is the time of obstacle detection by architecture, S_{stop} is the stopping distance for this architecture, V is the vehicle speed.

Table 4: Free computing time

Description	Free computing time, s
SSD Mobilenet v1	2.94
SSD Mobilenet v2	3.71
Faster R-CNN	2.24
YOLOv3	4.32

Experimental results demonstrated that variations of detecting network architecture in general case resulted in variations of safe speed at straight segment equaling to 15.32 km/h. At iced coated asphalt, the Faster R-CNN architecture provides safe speed of 36.19 km/h, which prevents application of additional algorithms of data analysis or control algorithms since it would violate project specifications to provide speed with upper limit of 25 km/h, thus making this architecture inapplicable for computer vision system of KamAZ BE.

The best result in terms of free computing time was demonstrated by YOLOv3, however, this was stipulated by higher distance of object detection (see Table 1) and moderate preset maximum speed of 25 km/h. The authors proposed procedure of reasonable selection of neural network architectures for object detection in order to provide various support for autonomous traffic of wheeled vehicles.

In addition, variation of detecting network increased stopping distance by 1.13 m and varied free computing time by 2.08 s.

IV. CONCLUSION

Embodiments of computer vision algorithms can effect significantly traffic properties of autonomous vehicle. Under actual conditions and upon entry of additional stages of image analysis to data processing pipeline, the delay between acquisition of raw data from sensors and acquisition of data ready for application in other systems or while accounting for delays in decision making system, the time delay before detection of dangerous situation, increases even more, which would exert critical influence on traffic safety of autonomous wheeled vehicle.

Additional studies of integrated computer vision system which performs multiprofile analysis of environment aiming at safe traffic in autonomous mode, are planned.

ACKNOWLEDGMENT

This work was supported by the Ministry of Education and Science of the Russian Federation, project 14.624.21.0049 (unique project identifier: RFMEFI62417X0049).

REFERENCES

1. A. Saykin, S. Buznikov, D. Endachev, K. Karpukhin, A. Terenchenko, “Development of Russian driverless electric vehicle”, *International Journal of Mechanical Engineering and Technology*, 8(12), 2017, pp. 955-965.
2. A. Saykin, S. Buznikov, K. Karpukhin, “The Analysis of Technical Vision Problems Typical for Driverless Vehicles”, Research Journal of Pharmaceutical, Biological and Chemical Sciences, 7(4), 2016, pp. 2053-2059.
3. Y. Tian, P. Luo, X. Wang, X. Tang, “Deep learning strong parts for pedestrian detection”, In IEEE International Conference on Computer Vision, 2015.
4. A. Saykin, S. Bakhmutov, A. Terenchenko, D. Endachev, K. Karpukhin, V. Zarubkin, “Tendency of Creation of "Driverles" Vehicles Abroad”, *Biosciences Biotechnology Research Asia*, 11, 2014, pp. 241-246.
5. S. Shadrin, “Affordable and efficient autonomous driving in allweather conditions”, FISITA World Automotive Congress, 2018.
6. COCO Dataset Common Objects in Context. Available: www.cocodataset.org/#detection-eval
7. J.C. Redmon, “YOLO: Real-Time Object Detection”. Available: https://pjreddie.com/darknet/yolo/
8. 8.Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, 2018. “Object detection with deep learning: A review,”arXiv preprint arXiv: 1807.05511.
9. S. Ren, K. He, R. Girshick, J. Sun, Faster, “R-CNN: Towards real-time object detection with region proposal networks”, In Advances in neural information processing systems, 2015, pp. 91–99.
10. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, “SSD: Single shot multibox detector”, European conference on computer vision, Springer, 2016.
11. A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, A. Hartwig, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”. CoRR abs/1704.04861, 2017.
12. M. Sandler, A.G. Howard, M. Zhu, A. Zhmoginov, L.C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks”, CVPR, 2018, pp. 4510-4520.
13. S. Shadrin, O. Varlamov, A. Ivanov, “Experimental Autonomous Road Vehicle with Logical Artificial Intelligence”, Journal of Advanced Transportation, 2017, pp. 1-10.
14. A. Ivanov, A.N. Narbut, A.S. Parshin, “Avtomobili: Teoriya ekspluatatsionnykh svoistv” [Automobiles: Theory of Operational Performances]: Guidebook. Academia, Moscow, 2013.
15. A. Ivanov, V. Gaevskiy, S. Kristalniy, N. Popov, S. Shadrin, V. Fomichev, “Adhesion Properties of Studded Tires Study”, *Jr. of Industrial Pollution Control*, 33(1), 2017, pp. 988-993.
16. A. M. Biniyazov, A. N. Bayakhov , A. Yu. Bektilevov , R. S. Sadykov , V. P. Zakharov & L. Kh. Sarsenbaeva, “Operation Maintaining of Automobile Forced Diesel Engines with Ensuring of Functional Condition of The Lubrication System In Exploitation”, *International Journal of Mechanical and Production Engineering Research and Development*, 9(3), 2019, pp. 1761-1768.