Ministry of Science and Higher Education

FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION OF HIGHER PROFESSIONAL EDUCATION

“Saint Petersburg State University
of Information Technologies,
Mechanics and Optics”

GRADUATION THESIS

Modeling the behavior of special transport based on the analysis of video surveillance data using computer vision methods

Author_______Grigorev Artur Konstantinovich_____ _____________
(full name) (signature)

Subject area ___01.04.02 Applied Mathematics and Computer Science_
(code, name of program track)

Degree level _______Master_____________________________________
(Bachelor, Master)*

Thesis supervisor Bochenina K.O., Assistant professor, PhD _________
(surname, initials, academic title, degree) (signature)

Approved for defense

Head of program __Ivanov S.V., Assistant professor, PhD__ _________
(surname, initials, academic title, degree) (signature)

“_____”__________________ 20 ____

St. Petersburg, 2019

Student___Grigorev A.K.____Group_____C4215c________ Faculty of__IDU_______________
(Surname, initials)

Subject area, program/major __01.04.02 Urban Supercomputing____________________________

Consultant(s):
a) ___Derevitskii Ivan Vladislavovich__________________________________ _____________
(surname, initials, academic title, degree) (signature)

b) ____ _____________
(surname, initials, academic title, degree) (signature)

Thesis received "____"_________________________20 ____

Originality of thesis: ______________%

Thesis completed with the grade: ________________________________

Date of defense "____"_________________________20 ____

Secretary of State Exam Commission_____________________________ __________________
(full name) (signature)

Number of pages _____________________________________

Number of supplementary materials/Blueprints __________________________________

Ministry of Science and Higher Education

FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION OF HIGHER PROFESSIONAL EDUCATION

**“Saint Petersburg State University
of Information Technologies,
Mechanics and Optics”**

APPROVED

Head of educational program

____________________ ___________
(Surname, initials) (signature)

«____» «_______________» 20____

OBJECTIVES

FOR A GRADUATION THESIS

Student___Grigorev A.K.__________**Group**___C4215c______ **Faculty of**________IDU_______

Thesis supervisor_Bochenina K.O., ITMO University, Assistant professor, PhD_______________

(Full name, academic title, degree, place of employment, position)

1 Title of thesis:__ Modeling the behavior of special transport based on the analysis of video surveillance data using computer vision methods____

Subject area ____Urban supercomputing _____________________

Program/Major_____01.04.02 Applied Mathematics and Computer Science _________________

Degree level_____Master____

2 Deadline for submission of complete thesis «______» «__________________» 20_____

3 Requirements and premise for the thesis ____ It is necessary to implement a method for detecting special behavior in video from surveillance camera; perform an analysis and justify the choice of computer vision methods applicable for the analysis of the traffic situation; collect data on road traffic parameters characteristic to the identified situations of special behavior; build a model of the behavior of a special transport based on the parameters of road traffic._________________________________

4 Content of the thesis (list of key issues)

_1. Identification of special behavior in video data using computer vision methods ._________

_2. Analysis of the parameters of road traffic, typical for situations of special behavior ._______

_3. Analysis and justification of the choice of methods applicable for the analysis of the traffic situation .____

4. Building a model of the behavior of a special transport based on road traffic parameters characteristic to the special behavior.____

5 List of graphic materials (with a list of required material)

6 Source materials and publications

7 Objectives issued on «____» «__________________» 20____

Thesis supervisor ____________________________
(signature)

Objectives assumed by ____________________ «____» «_______________» 20____
(signature)

Ministry of Science and Higher Education

FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION OF HIGHER PROFESSIONAL EDUCATION

**“Saint Petersburg State University
of Information Technologies,
Mechanics and Optics”**

SUMMARY

OF A GRADUATION THESIS

Student_______Grigorev Artur Konstantinovich____
(full name)

Title of the thesis_________ Modeling the behavior of special transport based on the analysis of___ video surveillance data using computer vision methods_______________________________________

____

Name of organization _______ITMO University____

DESCRIPTION OF THE GRADUATION THESIS

1 Research objective_ Identification, analysis and modeling of the behavior of special transport based on the analysis of video data using computer vision methods _________________________________

2 Research tasks__identification of special behavior cases in video data, analysis of parameters of the road situation, characteristic to the cases of special behavior using computer vision methods,_______ modeling the behavior of a special transport for the tasks of classifying a road situation and________ evaluating the possible acceleration from the implementation of special behavior________________

3 Number of sources listed in the review section __________67___________________

4 Total number of sources used in the thesis ________________74_________________

5 Sources by years:

Russian			**Foreign**		
In the last 5 years	5 to 10 years	More than 10 years	In the last 5 years	5 to 10 years	More than 10 years
1	0	0	36	21	16

6 Use of online (internet) resources ________________Yes, 2_____________________________
(Yes/No, number of items in the list of references)

7 Use of modern computer software suites and technologies (List which ones were used and for which section of the thesis)

Software suites and technologies	**Thesis section**
OpenCV, PyTorch, Keras, Pandas, Numpy, Matplotlib	3
OpenCV, Pandas, Numpy, Matplotlib	4

8 Short summary of results/conclusions ___25 cases of special behavior were identified, a classifying model was built, allowing to identify the situation characteristic of special behavior based on road parameters and a simulation model that allows to estimate the speedup from the implementation of special behavior. A new graphical feature TrackLoop and method for anomaly detection named Adaptive Suppression are proposed.____

9 Grants received while working on the thesis ____
(Grant name)

____

10 Have you produced any publications or conference reports on the topic of the thesis? _______
(Yes, no)

a) 1 _Grigorev A., Derevitskii I., Bochenina K. Analysis of special transport behavior using computer vision analysis of video from traffic cameras // Communications in Computer and Information Science - 2018, Vol. 858, pp. 289-301 ____

(Bibliographical description of a publication)

b) 1__Presentation of paper “Analysis of special transport behavior using computer vision analysis of video from traffic cameras” on 3th International Conference Digital Transformation & Global Society (DTGS 2018) __

(Bibliographical description of a conference report)

2__ Report on topic "Identifying special behavior using computer vision analysis of video from traffic cameras" on VII Congress of Young Scientists (VII Конгресс молодых ученых), 2018.

Student__________Grigorev Artur Konstantinovich________________________ ______________
(Full name) (signature)

Thesis supervisor____Bochenina Klavdiya Olegovna______________________ _______________
(Full name) (signature)

“__________”________________20___

TABLE OF CONTENT

INTRODUCTION

The transport network is an important part of the city’s infrastructure. and is a system - the sum of the interacting parts, which are arranged and operate according to specified rules (traffic rules, building codes and regulations defining the structure and connection of roads). One of the elements of the transport network is a transport that performs special tasks - going to the scene of the accident (police cars), transporting patients (ambulance cars), reaching the scene of a man-made accident to eliminate it (fire service cars, emergency ministry vehicles, etc).

Special transport (e.g. ambulances, police cars) has a special of possibility to bypass some of the traffic rules (e.g. ignore traffic lights, move by the oncoming lane) when necessary. Thus, it can travel noticeably faster and avoid traffic jams. Modern traffic models analyze the movement of special transport without consideration of this possibility. Common decision support systems (e.g. GPS-navigators, route-planning systems) give only rules-based recommendations (without considering the possibility to bypass them). To develop a decision-support system for special transport, it is necessary to develop specific models. To develop special behavior models, it is necessary to collect the data about patterns of special vehicle behavior and analyze traffic conditions related to the cases.

Optimizing the operation of special transport is an important task, the solution of which can improve the quality of life of people through the timely elimination of accidents, early assistance and prevention of accidents. Relevance of the research is stated by the necessity to reduce travel time in emergency situations (e.g. patient transportation, fire truck mission).

For large cities, large travel distances are characteristic, and for cities with high population density, congestion is typical. These two factors negatively affect the speed of movement of special vehicles, for which the time of movement is critical. The traffic rules are defined by the unique possibilities of special transport - ignoring certain traffic rules:

1. Move exceeding speed limits;
2. Ignoring traffic lights;
3. The ability to move by the opposite lane (MBOL);

The decision on the application of these opportunities is made by the driver situationally and momentarily (since the outcome of the move will depend on the time spent on decision-making). Decisions require a certain accuracy of information that modern services either provide to a limited extent (averaged parameters with hour accuracy) or do not provide (for example, the presence of cars in the opposite lane).

Recent advances in computer vision allow to analyze the traffic situation with high accuracy, which allows to build a model of special behavior. Using the model, it is possible to make decisions regarding the implementation of special behavior on any of the observed road segments, with certain parameters of the road situation.

The purpose of this work is to identify special behavior, analyze and build a model of special transport behavior based on video data from the St. Petersburg camera, using computer vision methods, and propose new methods to analyze the behavior of special transport.

Another purpose of this work is to develop an automatic system for the classification of road situations, which are characterized by special behavior, based on the parameters of the road situation collected by computer vision methods.

The tasks of the thesis are:

1. Research into existing methods and technologies of computer vision, providing detection and tracking of objects, counting objects in the image, semantic segmentation, optical flow analysis methods for subsequent analysis of the parameters of the traffic situation, corresponding special behavior situations;
2. Development of the system using the considered computer vision methods for detecting and analyzing cases of special behavior;
3. Construction of a model of special behavior;

The theoretical significance of the work lies in the study existing methods (including deep learning methods) in computer vision tasks applicable to the analysis of road traffic. Applied significance lies in the design and development of a system to demonstrate the feasibility of these methods.

The object of the research is the methods of computer vision that can collect data on the behavior of special transport and the parameters of the road situation that are relevant to the situations of manifestation of this behavior.

The subject of the research is the combination and adaptation of these methods to the task of recognizing special behavior and analyzing the parameters of the corresponding traffic situation.

The traffic parameters obtained in the course of modeling applied to multi-agent modeling of road segment, considering the movement of special transport, to assess the benefits of implementing the rules of special behavior.

Classification model (which can help to decide when MBOL should be performed) and simulation model (to assess possible speedup, considering real traffic parameters obtained using computer vision) are made.

Scientific novelty of the work:

1. A novel graphical feature “TrackLoop” for description of dynamics in video.
2. A speed-density decision model for the MBOL-behavior of special transport.
3. Anomaly detection method named “Adaptive Suppression”, which relies on distance (similarity) metrics only.

MAIN PART

1 SPECIAL TRANSPORT RESEARCH REVIEW

Important case, when the travel time of a special transport is critical, is the stroke. According to [1], stroke kills about 140,000 Americans each year and is related to 5% of all deaths. Within one hour of stroke, a patient loses as many neurons as he usually loses in 3.6 years[2].

The time of delivery of a patient is known to strongly affect the benefit of intravenous thrombolytic therapy and the probability of success of recanalization therapy[3] if the patient arrives within 60 minutes - the so-called "golden hour". Suddenly, from 253,148 ischemic stroke patients, only 28.3% arrived within one hour to a hospital. Chances of a positive clinical outcome decrease by 38% resulting from every additional hour in travel duration, even with age and reperfusion adjustment[4]. Travel time to scene for ambulance vehicle greater than 20 minutes is known to be associated with higher odds of mortality than travel time less than 10 minutes[5].

According to [6], the traffic conditions on the roads have a significant impact on the ambulance response times. Therefore, a research relating to the reduction of the travel time of a special vehicle is associated with preserving life, reducing the damage received from diseases, and improving the quality of therapy for large number of people.

Special transport travel time can be reduced as a result of ignoring traffic rules. But special behavior occurs not always. Thus, we need to analyze road situations related to cases of special behavior, as well as detect such cases.

The past decade has seen the rapid development of computer vision methods. Recently, a considerable literature has grown up around the topic of computer vision methods and related traffic analysis systems. Applicability of computer vision methods can be infered through comparison with existing sources of traffic data (e.g. GPS).

The one of objectives of the review is to determine which metrics are appropriate for the traffic analysis task, different traffic state measures, as well as the computer vision methods that allow collecting such data. Computer vision methods can do the classification of detected objects. Consequently, it can allow to distinguish special transport among other vehicles.

1.1 Ambulance routing and distribution

To schedule an ambulance route, we can consider number of vehicles, traffic congestion, static of dynamic nature of events in the transport system. Route can be calculated for each vehicle independently or for an emergency vehicles fleet. Route planning can be static, using road segment cost metric[7], or dynamic, considering change in traffic congestion during the day[8].

Distribution of the special transport in the city is another field of research, where consideration of special behavior can be used. It is possible to use a static planned vehicle distribution in the city, minimizing the average time of the car's journey to the place of call, or to carry dynamic relocation of idle vehicles (in order to maximize number of calls reached by an ambulance service)[9,10], depending on the routing cost.

To predict the arrival time, a model based on real measurements of traffic parameters on road sections can be built. However, these models do not consider the possibility of special traffic to use special behavior. Most studies in the field of ambulance routing have only focused on generic vehicle route planning.

One of the most significant current papers on the topic of special behavior is [11], where author takes special behavior into account and compares the two methods of route planning - with and without possibility of special behavior. Author presented a model of special vehicle behavior, considering traffic density and number of lanes:

1. The author focuses on a single ambulance response to a single incident. Therefore, modelling a multitude of calls and a set of vehicles on a real city map and using real parameters is the subject of the task of further modelling to assess the advantages of special behavior.
2. The necessity of parameterization of the received transport model, considering the possibilities of special transport by collecting data on the

behavior of a statistically significant number of ambulances, is also indicated.

3. The time of the emergency travel is compared with and without consideration for the ability to violate the rules.

This is the main article to which this study is addressed. The use of computer vision methods provides the possibility to estimate the parameters of motion patterns of special transport with data on the state of traffic for cases of special behavior.

It is known that patient arrival time affects the choice of the therapy as noted in [12], but since the special behavior is the behavior of the micro model, the effect produced by the consideration of special behavior on the whole healthcare is not yet known. However, author proposed analytical approach (special transport always travels by the oncoming lane getting different speedup (derived analytically) depending on road segment traffic flow), not a dynamic logic of an agent in multi-agent simulation (with the scope of road or road network). Thus, opening the field for further research.

The route planning system can be supplemented with road intersection models and collected the data on the state of traffic, which should ideally lead to a green wave (an ecological term meaning the absence of movement inhibition) for special transport.

Result of route planning, on a randomly generated traffic system, shows significant decrease in travel duration (up to 5 times less). But author also states, that it is necessary to parameterize the model using real traffic data (thus, it is necessary to collect it). Collected parameters will allow performing further simulations related to the specific traffic system of the city.

Special vehicle driver does not always decide to move by the oncoming lane. To find traffic conditions under which this situation happens, it is necessary to analyze real traffic data.

The further study on the topic of existing planning models, location and relocation of special transport in the city, considering the possibility of special behavior, is undiscovered field of research. Dynamic route planning, depending on the current traffic congestion, considering special behavior, has potential to produce completely different routes, in comparison with static route planning for generic vehicles, therefore reducing emergency services travel time.

In [13] authors present a real-time emergency vehicle redeployment method using urgency parameter of received calls. The research can also be useful for planning redeployment of emergency services in accordance with the degree of importance of calls with consideration of possibility of special behavior. In [14] authors studied on the literature of the effective healthcare facility location.

Taken together, these studies support the notion that research on the decrease in travel time of emergency vehicles is significant.

Thus, consideration of special vehicle behavior can affect multiple aspects of emergency services route planning.

Until recently, there has been little interest in special vehicle behavior. But, location and re-deployment planning, planning of static and dynamic routes, ambulances distribution in the city and even the distribution of the healthcare buildings, the entire system of patient care and patient delivery can be deeply affected using models of special transport behavior, which can be used in decision support systems (travel navigators and route planning systems), which can give recommendation considering possibility of special behavior.

The development of models for the movement of special vehicle can use different levels of detail. Micro models of special transport can be represented as agent model on the road, and macro models can be represented as decision model considering statistical parameters).

1.2 Collection of the data

In [15] it is mentioned that a single-frequency band GPS has error about 10 meters and in [16] it is stated that GPS-enabled smart phones are usually accurate to within a 4.9 meter radius under open sky (with worse accuracy near buildings), whereas the width of the roadway is 3.75 meters, according to Russian building codes and regulations.

Therefore, the usual GPS (installed in cars and smart phones) does not allow to determine the position of the car with lane accuracy (for example, in the case of 4 lanes), and, accordingly, detect movement to the oncoming lane. The method suggested by the authors in [17] markedly decreases the GPS measurement error by 20% by adjusting the GPS values by the values of the inertial sensors, but this is still not enough to distinguish the lanes.

In [18] authors proposed a methodology for collecting data on cases of travelling to the oncoming lane. However, they analyzed a short period of time and identified a few cases of travelling to the oncoming lane. No attempt was made to collect the data on traffic flow or traffic density at the time of such cases. Only one of the computer vision methods for traffic analysis is used, while many others can be compared and the most productive and suitable for the task can be selected. Also, the study fails to consider the facts of driving to the red light and exceeding the speed limits are not analyzed.

GPS is valuable source of the data. However, it is necessary to use more precise methods, e.g. computer vision analysis of video. Its accuracy allows to analyze traffic state within lane [19].

The analysis of traffic by computer vision is usually not limited to one method and includes the region of interest (ROI) selection, detection and tracking of objects on video, camera calibration, etc. To implement the detection of cases of special behavior, we can use existing combinations of computer vision methods.

In [20] introduced Ambulance Driver Questionnaires measuring aspects of driving performance, style and driving competence. Main driving performance problems were bad visibility and misjudgement of the distance to the oncoming vehicle. These problems can be addressed by the integration of computer vision road analysis and decision support system (GPS-navigator) to supply or, in some cases, replace the limited driver's vision and provide accurate distance measurement.

1.3 Complex traffic surveillance systems

Modern traffic surveillance systems allow analysis of traffic situation. As a rule, complex systems are used in the traffic analysis. Traffic surveillance system implies high-performance computations on video, high-performance network of data-sources (traffic cameras) and data-storages as well as computation devices. This type of system develops in the direction of fully automatic calibration[21] and automatic highlighting of interesting events, providing the user with only important data for decision making and not forcing him to monitor the entire video.

The system presented in [22] implies the use of separate servers for feature extraction, querying video data by parameters (with filtering by the type of observable activity of subjects, for example, you can request all the video fragments where fights occur, or more precisely by subject's actions).

2 COMPUTER VISION METHODS IN TRAFFIC ANALYSIS

2.1 Traffic counting

Baseline (detection line) method[23] is the use of virtual line that cuts through the road, intended to detect objects that intersect the line. Line has two states: “occupied” and “released”. Switch between states increases the counter of traffic flow. To detect occupancy, usually background subtraction method is used. It allows to represent object which distinct from background as white blobs. Thus, counting white pixels on extracted detection line, we can say about occupancy of object.

Baseline method is suitable for freeways and intended to be used for high speed moving vehicle counting with noticeable distance between cars. But in traffic jam, cars are close to each other (partial occlusion happens) making background subtraction method produce connected blobs. This method is not suitable for roads with traffic congestion. There is also adaptation of this method to three baselines – start, middle and end, connected using simple logic: when first two lines occupied – raise flag, when flag raised, and last two lines are occupied - increase counter. It can be said that logic can be not only strict but fuzzy and detection lines can be used as a feature in Deep Learning when count logic is not yet defined.

Virtual loop[24] method is the use of rectangle shaped detector that placed on a lane exit. Presence of object determined by the occupied area of a virtual loop. It has two states as in baseline method. Occupancy also detected using background subtraction and represented as white pixels in the virtual loop area. If occupied area is above the given threshold, state of a virtual loop becomes “occupied”. Switch between states increments vehicle counter. Method can be used on roads with traffic congestion. But still can suffer from disadvantages of background subtraction (need of shadow removal, necessity of specific view angle which reduces partial occlusion of vehicles).

There is also implementation of adaptive vehicle counting system, which proposes switch between different methods depending on traffic state[25]. Thus, eliminating disadvantages of methods in certain traffic states.

2.2 Traffic speed

Average road/lane traffic speed can be estimated through optical flow computation. There are two types of optical flow computation methods:

1) Sparse optical flow computation – optical flow computed only for multiple local areas, which usually contain graphical features (e.g. corners, which can be easily tracked).
2) Dense optical flow computation – optical flow is computed for the whole image, for every pixel. Redundant, but can capture high speed of movement and produce graphical interpretable maps of movement magnitude and angle.

In [26] authors use Lukas-Kanade (LK) optical flow method to track the movement of features on the image and determine the speed of traffic, but also using projective mapping to project perspective image of the road on rectangular plane, making speed of vehicle less distorted.

Gunnar-Farneback [27] method used to calculate dense optical flow. The method calculates oriented three-dimensional tensors in a sequence of images that are combined within a parametric model to estimate the speed. This method estimates the speed of points in the region, and not for each point separately, which produces more smoothed optical flow maps. The method can be applied over images of different scale, capturing tensors and combining them subsequently. Thus, it is possible to obtain an optical flow related to objects, and not just over certain parts of objects.

Macroscopic optical flow velocity(MOFV) analysis - is an estimation of the whole traffic flow speed[28]. Method is based on distribution of apparent movement velocities of brightness patterns in sequential images. Horn algorithm used for calculating the motion vectors on a large scale and Lucas-Kanade algorithm used to capture local movement.

2.3 Traffic density

Traffic density is related to density of vehicles that occupy the road segment at the moment in time.

Road space occupancy is a percentage of used road area. Traffic density can be estimated through road space occupancy. With some loss of precision, in the situations, when difficult to distinguish the individual vehicle in surveillance video, we can use the density of image features (e.g. edges) to substitute the density of vehicles [29]. Well-known computer vision methods such as background subtraction[30] or semantic segmentation[31] can produce segmented area of pixels, which can be used to estimate road occupancy.

For simplicity, edge detection algorithms can also be used to approximate road occupancy[29]. Sobel algorithm – simple edge detection convolution can be used to extract the vehicle edge map in the road areas. It also less affected to the illumination change. Then after binarization, road space occupancy calculated as ratio between edge pixels present in binary image of edges and the number of pixels in the road. Derived value, calculated for each lane, can be used to compare traffic density between lanes.

2.4 Application of segmentation methods for traffic density estimation

In this research, to estimate the traffic density, it was decided to use the segmentation method based on convolutional neural networks. The autoencoder U-Net architecture [31] was chosen as a convolutional neural network (Figure 1). U-Net is a symmetrical convolutional autoencoder with symmetric skip connections between layers.

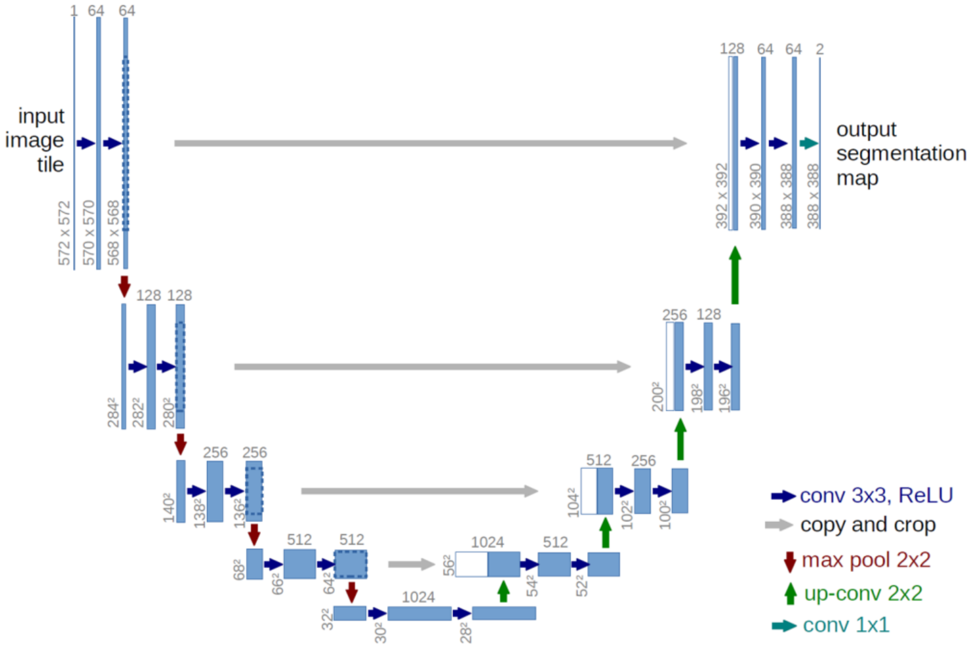

Figure 1 — U-Net architecture

Source: O. Ronneberger. U-Net: Convolutional Networks for Biomedical Image Segmentation.

Architectures similar to U-Net has already been used to estimate traffic density [32], estimate traffic count or count people in the crowd using block regression from a density[33] or block regress count value directly from an image[34]. Before U-Net, full-convolutional neural networks were used to count microscopic biological objects[35].

Marked data (Figure 2) is used to train the autoencoder, which are used as the source image and segmentation map, respectively.

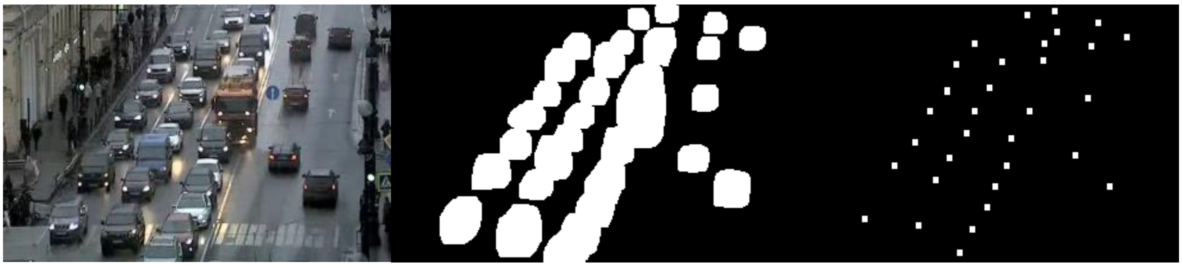

Figure 2 - Frame example (left) and manual marking for segmentation (center) and counting (right)

The U-Net architecture is based on a full-convolutional neural network but has better segmentation quality. This is achieved by using skip connections between the upsampling and downsampling layers using the concatenation operator. This allows the neural network to combine raw data and segmentation results at different levels of resolution.

As a metric for evaluating the performance of this neural network, the Sorensen-Dies coefficient can be used:

$$\mathrm{dice}(X, Y) = \frac{2X \cap Y}{X + Y} \qquad (1)$$

Formula 1 means that the metric is equal to the doubling of the number of elements common to both sets, divided by the sum of the number of elements in each set. Also, the Sorensen-Dice metric is known as the F1 metric. In terms of segmentation, X is the result of a neural network, Y is a segmentation map (obtained, for example, as a result of manual markup).

The convolutional neural network has a unique feature to perform segmentation not only along the contour of an object, but to reduce segments on a segmentation map to segments of a certain shape and size. In particular, one can use small segments (noticeably smaller than the initial segment), which allows you to segment objects in the image as points (Figure 3). This, further, allows to count objects. In this work, also, an attempt was made to count objects using the U-Net architecture.

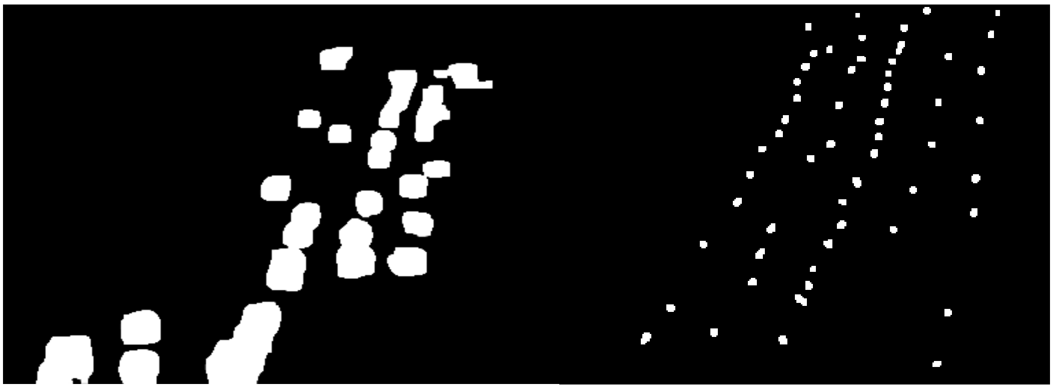

Figure 3 - Demonstration of the use of segmentation to segments (left) and segmentation to points (right)

2.5 Fast neural network architectures for semantic segmentation

A convolutional neural network can perform segmentation to segments of a certain shape and size. Segmentation to points is often used to count various objects, including the counting of microscopic biological and human objects in a crowd — medical segmentation and crowd counting — these two are not only ways to use segmentation, but important areas of research in the field of computer vision segmentation methods with its own requirements and approaches.

The semantic segmentation method is applicable for estimating both the density [4] and the amount of traffic (when segmenting to a point). As far as is known, point segmentation has not yet been applied to counting machines in a scene. This type of segmentation provided for vehicles in current research to perform the counting of transport objects at a viewing angle that causes partial occlusion. Partial occlusion can produce up to 90% occlusion of an object. Ability of convolutional networks to use multiple learned filters can improve recognition of specific patterns related to partially occluded vehicle, making it possible to detect in highly congested scene. Segmentation to point can produce point map of partially occluded vehicles, making it possible to count them.

SegNet [36], proposed in 2015, is an example of one of the earliest architectures for semantic segmentation. However, the number of parameters of this neural network is about 40 million. Modern architectures (including architectures that achieve image processing speeds of 30 frames per second or more) for semantic segmentation have a much smaller number of parameters. Less number of parameters allows to compute faster with less memory requirements, making it possible to process Big Data or video in real time.

Number of parameters for modern semantic segmentation networks when processing Cityscapes [37] dataset:

1. ERFNet - 2.1 million parameters.

2. ENet - 0.36 million parameters.

3. ContextNet - 0.85 million parameters.

4. EDANet – 0.68 million parameters.

With a much smaller number of parameters (for comparison, EDANet has 83 times fewer parameters than SegNet), new architectures provide better accuracy and processing speed[37]. SegNet, unlike EDANet or BiSeNet, is a symmetric architecture (as well as U-Net). It is this quality that makes the used neural network decoder a resource-intensive element. EDANet and BiSeNet, relying on asymmetric architectures, process the feature space, completing the transformation with a scaling layer. For this reason, some small details can be ignored in the learning process of a neural network, because, for example, a reduced 8 times image does not contain some small details that would be important for accurate segmentation.

Another metric used to estimate quality of segmentation is IoU metric. The methods of semantic segmentation are compared by the Intersection-over-Union metric[38], also known as the Jacquard index, used to compare graphical objects of arbitrary shape. A distinctive feature of the metric is the scale invariance, since the objects are compared by the occupied graphic regions and the subsequent calculation of the normalized measure associated with the areas of the objects. However, the metric is not representative if the graphical objects are far away ($| A \cap B | = 0$, $IoU (A, B) = 0$). Also, in the problem to be solved, there are only two classes of the object — the machine and not the machine; therefore, it was decided to use the Sorensen-Dice metric.

U-Net and SegNet are examples of symmetric auto-encoder architectures based on full-convolutional neural networks. Due to symmetry, the decoder block has the same number of parameters as the encoder, which explains a large number of parameters. It makes these architectures slow, hardly applicable in practice.

EDANet (Efficient Dense modules with Asymmetric convolution) is one of the new architectures for semantic segmentation. A distinctive feature is high speed (3 times faster than ICNet): this neural network is capable of processing high-resolution images at a frequency of 108 frames per second with mIoU = 67.3% [37].

Using the asymmetric non-decoder architecture EDANet achieves high performance.

The EDANet architecture (Figure 4) consists of:

1. Two dimension-reduction units

2. Two EDA blocks (each EDA block contains dot convolutions (with the size of 1x1 cores) at the input and output of the block, as well as sequentially located asymmetric convolutions (1x3.3x1) inside the block).

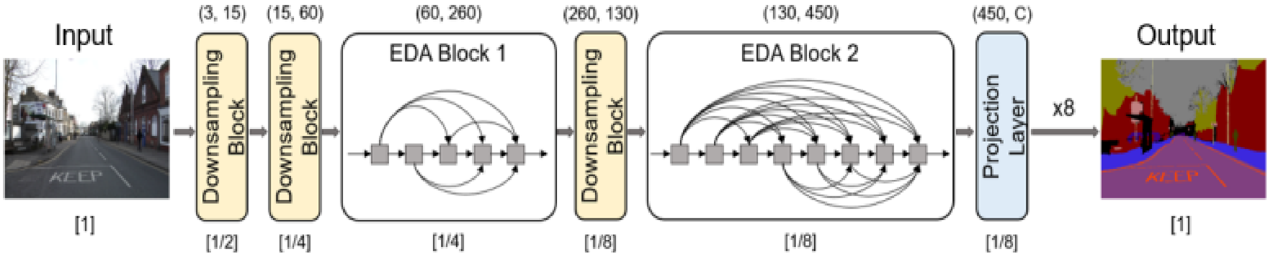

Figure 4 - The EDANet neural network architecture

Source: Lo, Shao-Yuan, Hsueh-Ming Hang, Sheng-Wei Chan, and Jing-Jhih Lin. For real-time semantic segmentation

Pointwise convolutions. Point convolutions are an element of Inception-architectures and a filter with 1x1 kernel, which performs convolution on pixels in depth.

Point convolutions used to manipulate number of channels and can significantly reduce the computational complexity of the architecture by reducing the number of channels.

Assymetric convolutions. Asymmetric convolutions are the solution to substitute an N x N convolution into two consecutive convolutions of N x 1 and 1 x N [39], allowing the convolution to be calculated by sequential memory access when calculating the convolution. resources.

Dilated Convolutions (Astrous Convolution) also reduce the need for computational resources by exponential expansion of the receptive field of convolution and a dilation between convoluted pixel blocks. Dilated Convolutions are used in the second asymmetric convolutions block in the EDA block.

Another example of a fast segmentation architecture used in this work is the BiSeNet (Bilateral Segmentation Network) [40]. In this architecture (Figure 5), high performance is achieved due to the parallel calculation of the spatial and contextual path (which is needed to derive a large receptive field). The spatial path is to consistently apply convolutional layers to obtain 1/8 dimension for obtaining detailed image features. The context path uses Global Average Pooling based on the attributes of a pre-trained Xception architecture (which provides a quick dimension reduction) and thus context calculation. The context path uses Attention Refinement modules. Then, to display the final segmentation map, the two paths are combined using the Feature Fusion module.

Attention Refinement module is a special module consisting of layers of global pooling, point convolution, a packet normalization layer and a sigmoid activation function and a parallel path that multiply. The module is needed for the processing of signs at each stage.

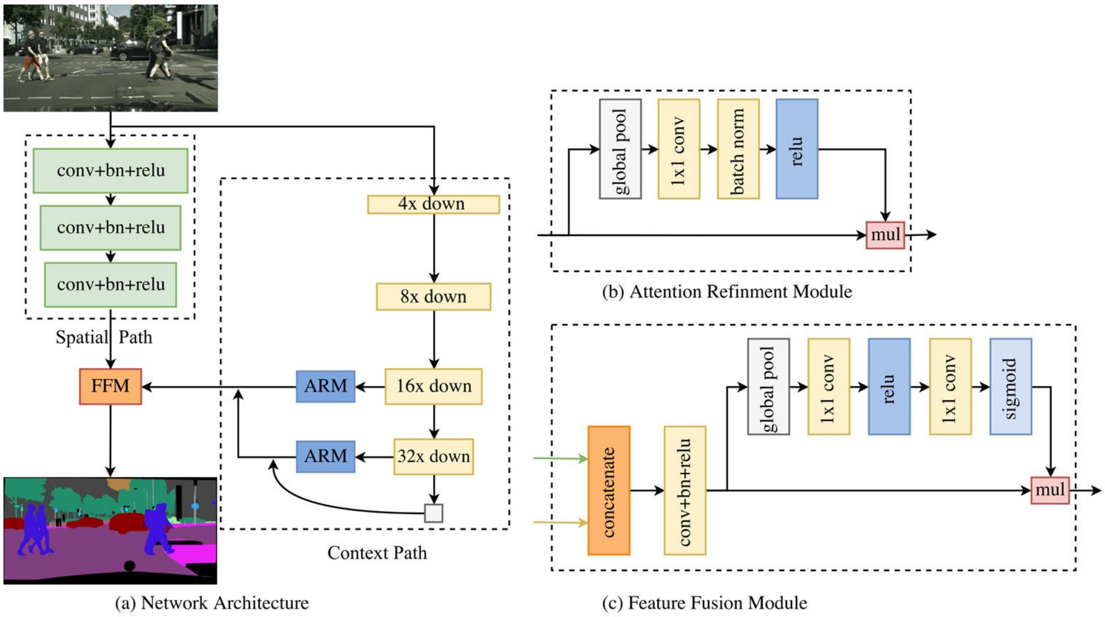

Figure 5 - Architecture of the BiSeNet bilateral segmentation network

Source: Yu, Changqian, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation.

2.6 Vehicle detection

Viola-Jones cascade of classifiers - a machine learning approach for visual object detection, which is capable of rapid processing of images[41]. Viola and Jones's AdaBoost learning algorithm usually utilized for the selection of small number of appropriate features (Haar wavelets) and construction of binary classifiers. Binary classifiers organized into cascade that provides statistical guarantees that discarded regions are unlikely to contain the object of interest. Haar features have proven to focus on what remains constant over different detection windows with small variations (viewpoint, lighting). Method is sensitive to illumination, weather conditions, perspective distortion and rotation. Cars, depending on road camera point of view, are rigid in form, therefore this method is applicable and could produce sufficient results. Because of sensitiveness a lot of different positive and negative images need to be gathered. It is possible to combine multiple detectors (e.g. detectors that based on front and side view images of vehicles).

The method achieves high performance with the use of 1) a cascade of simplest classifiers, which allow ignoring certain areas of the image with a high probability that do not contain the desired object 2) using the integral sum when calculating Haar wavelets. The Haar wavelet is a graphic feature defined by a shape with white and black areas. To calculate the compliance of the image with the characteristic, the calculation of the sums of pixels in the white and black areas is performed. The ratio of white to black pixels for this part of the image is used as a value for classification.

When constructing a cascade, at each stage, the choice of a feature is performed that provides the best classification quality.

Histogram of oriented gradients(HOG) is a method based on classification of histograms of oriented gradients descriptors[42]. Image window is divided into small spatial regions("cells"), for each cell accumulating a local 1-D histogram of gradient directions (HOG descriptors) or edge orientations over the pixels of the cell. HOG descriptors then need to be classified with methods like SVM (support-vector machine). HOG outperforms wavelet and SIFT methods in pedestrian detection and also method can detect cars of different color under different lighting conditions (day, night width road lights, morning, cloudy/sunny day) due to low sensitivity to intensity

Background subtraction(BGS) – is an object segmentation method, that based on extraction of pixels that different from the background (usually there is a threshold of difference given)[43]. BGS is one of the easiest methods for extracting scene objects.

BGS usually has many sources of errors (e.g. shadow, partial object occlusion), but also it is the richest method in terms of quantity of approaches to eliminate this errors.

BGS method consist of two stages:

1. Background learning – background image learned using statistical methods (such as Mixture of Gaussian Models[44,45]).
2. Background subtraction – difference with learned statistical model of image sequence is produced for image.

Both stages can be executed simultaneously in real-time background learning.

There are different types of errors, that can happen when using this method for car segmentation (like car occlusion and shadow casting). In this case, multiple vehicles in the occlusion can be identified as one object. Car windshield usually has small difference with background(road) color. Therefore, it is almost impossible to get solid blobs for cars as a frame-background difference.

Background subtraction can effectively be used in traffic monitoring for freeways[46], but are most likely to fail when car occlusion happens. Thus, background subtraction is not desirable to use for highly congested roads analysis with camera angle which implies frequent observation of partial vehicle occlusion.

2.7 Vehicle tracking

There are two most used object tracking methods, that based on histogram representation of objects [28]:

- Mean-shift – method based on plotting of a 2D probability space where an object template can be located with the help of corresponding template matching algorithm that aims to locate the maximum in a 2D probability space in order to specify the location of a predefined template. Method is weak on perspective images;
- Particle filters– a path estimation method that recursively predicts the target object’s posterior with discrete sample-weight pairs in a dynamic Bayesian framework. This method is able to cope with non-linear non-Gaussian assumption.

These trackers combined with visual representation unable to cope with low framerate of video, poor motion continuity and rapid changes in the appearance[47]. Also, occlusion should be mentioned because of change in the visual representation of an object.

Methods are robust in tracking of rigid templates but perform poorly with object tracking in perspective video. To solve this issue mean-shift method has been improved to CamShift (continuously adaptive mean-shift) which adaptively adjusts the tracking window’s size, but CamShift algorithm performs poorly when the object to be tracked suddenly gets occluded by an obstacle [48].

The movement of the vehicle is characterized by a gradual change in the motion parameters (speed, acceleration) and direction. To track such objects, the MedianFlow algorithm can be used, based on the path prediction error [49]. The MedianFlow method for tracking objects uses trajectory prediction not only forward, but also backward in time and compares the error in the difference in trajectories. The method uses the assumption that the correct tracking should be independent of the direction of time. First, the object is tracked forward in time and a straight trajectory is built. If the trajectories are very different, the straight trajectory is discarded. Distinctive feature of the method is the ability to detect facts of incorrect tracking.

Kalman filter can be used to predict the trajectory of a vehicle in consequent frames. The Kalman filter point tracking is the most popular among the parametric approaches, which obtains an analytical solution for tracking by assuming linear dynamics of vehicular movements and a Gaussian distributed displacements of vehicular objects [50]. When tracking is performed, an algorithm searches for possible image displacement in nearby area. And the task of a filter is to reduce number of comparisons using the probability of next point displacement based on its previous movement statistics.

Kalman filtering nevertheless can produce inaccurate results when estimating possible movement from image due to significant perspective distortion (change in planar image coordinates has different relation to distance and depends on a view angle and therefore significantly differs between near and far road segments). It is possible to improve movement prediction using projective transformation.

2.8 Traffic video anomaly detection methods

The behavior of a special vehicle is abnormal for a traffic situation. Therefore, to identify this type of behavior, you can use various existing methods of machine learning to identify anomalies. Also within the framework of the work, a new method of detecting anomalies is proposed.

The method in [51] uses the background subtraction technique and tracing the resulting points of interest using the KLT method to obtain ROI. To account for the graphical anomalies histogram of oriented gradients (HOG) is used. To take into account dynamic anomalies, it was proposed to use a histogram of oriented swarm accelerations, which is an agent-based model of “predators” and “victims”, based on tracking methods, which takes into account inertia and interaction of “predators”. Since the method relies on the analysis of areas (both in terms of appearance and in terms of dynamics) and two types of descriptors, the method can distinguish areas of visual, dynamic and visual-dynamic anomalies. There is also a joint use of visual and dynamic features - using HOG and HOF (optical flow histogram)[52].

Hierarchical clustering of trajectories based on similarity can also be used[53]. The method, due to the analysis of trajectories, can distinguish anomalous trajectories of movement of objects. However, trajectory tracking makes the method susceptible to losing track error.

The most popular feature for the problem of isolating anomalies is a mixture of dynamic textures [54–56] - a generative model that models a set of video sequences derived from a dynamic texture. Each space-time block is defined by a linear dynamic system in which the subsequent hidden state is determined based on the vectors of the previous hidden state and the process vector, and the representation vector is determined through the vector of the previous hidden state and the noise vector [57].

Among other feature used to detect anomalies [58]:

1. Feature vector extracted using convolutional neural network.
2. Fisher trajectory vector.
3. Code words of space-time regions.
4. Spatio-temporal cuboids [59].

2.9 Anomaly detection techniques

In this work, a comparison was made of the Isolation Forest and OneClass SVM anomaly detection methods applied to the dynamics feature. It is assumed that after training the methods on video with normal motion, the methods will determine the movement in the oncoming lane as a significant anomaly in the video with movement on the oncoming lane. Isolation Forest and OneClass SVM are unsupervised methods that analyze the distribution of points.

The Isolation Forest method is a method whose task is to isolate anomalies using trees, and not to extract norm maps [60]. According to the authors of the method, methods relying on the construction of profiles of normal values are not optimized to detect anomalies directly, giving rise to many false positives. The advantage of the method is that it is not limited by the dimension of space and is fast. Anomalies are isolated closer to the root of the tree, because they are rare and far from the basic data (that is, it is easiest to isolate from the rest of the data). Anomaly is determined by the depth of isolation in the tree.

OneClass SVM is a method based on non-linear mapping of features into multidimensional space using kernels. The rule-oriented method will look at how many points fall into a certain region and, based on this, draw a conclusion about the norm in a region. OneClass SVM works in the opposite way: it starts with points that must fall into a region and then calculates a region that has the desired property. Initially, many such regions appear. Therefore, regularization is used to reduce the region associated with the core. OneClass SVM, can be used to identify anomalies in the video when using space-time cuboids [61]. Also, OneClass SVM can be applied to the semantic vector resulting from the training of a convolutional neural network. However, the semantic vector contains instant data related to a particular frame. To correctly determine the anomalies of the dynamics itself, it is necessary that the code vector of the semantic neural network contains data on the dynamics in a time sequence.

The anomaly detection method can also be based on the background subtraction method used by the Gaussian Mixture Model [56]. When using this method, a map of distributions is constructed based on a weighted Gaussian sum for each pixel of the image. GMM parameters are calculated based on the Expectation-Maximization iterative algorithm. All that is outside the resulting distribution is an anomaly (or, as in the background subtraction method, an object other than the background).

2.10 Dimensionality reduction techniques

PCA (principal component method) is a statistical method that uses orthogonal transformation to transform multidimensional data into a set of linearly uncorrelated variables (components). Each resulting component is a linear combination of input features. Components have order. The very first component has the greatest variance. The method was proposed in 1901 by Pearson [62].

TSNE (t-distributed Stochastic Neighbor Embedding) [63] is a multidimensional data visualization tool (proposed in 2008) that converts the similarity between data points into conditional probabilities and tries to minimize Kullback-Leibler divergence between conditional probabilities of a small dimension and a larger dimension space . TSNE has a nonconvex loss function (depending on the initial conditions, different data representations can be obtained).

UMAP (Uniform Manifold Approximation and Projection) [64] is a new dimension reduction method (proposed in December 2018), based on the use of Riemann geometry and algebraic topology. UMAP is comparable to TSNE in visual quality and has better performance. UMAP optimizes the placement of data in a small dimension space to minimize cross-entropy between two topological representations. The disadvantage of the method is its non-interpretability. Also, UMAP is based on the axiom that local distances are more important than global distances. Therefore, UMAP is not recommended for capturing a global structure.

2.11 Traffic video preprocessing

Video preprocessing techniques used to reduce computation time, requirements for computational resources, requirements for network bandwidth, etc.

Road area extraction is an extraction of pixels that belong to the road. It is used to reduce false positives in detection algorithms, that can happen especially when using feature-based detection and background subtraction methods used. There are different approaches including frame difference accumulation [65] and region of interest (ROI) definition, usually represented as geometrical figure.

Frame difference accumulation is the accumulation of frame difference over video frames. After accumulation, each accumulated pixel value divided by the number of frames. Then binarized image(mask) based on average value excess produced. After mask generation, erosion and dilation operations can be used to reduce noise both in black and white areas and to expand mask to regions that was not detected as a road area (because of rare pixel change) but has high probability to belong to road.

2.12 Camera calibration

Camera calibration(projective mapping) is the process of estimating camera parameters so that pixel points in camera coordinates can be mapped into real-world coordinates[66].

In the literature, the term “Camera calibration” tends to be used to refer to mapping of objects position on an image to the real-world physical or virtual space coordinates. Calibration of the camera can be done through evaluation of camera parameters - focal length, pan angle, tilt angle, height of camera.

In practice, it is required to determine the coordinates of the vehicle on the road and its speed in physical units (e.g. meters, km/h). To display the coordinates of screen points on a virtual geometric plane, projective mapping (projective transformation) is used, which consists in defining a special transformation matrix, to which the vector of coordinates of the object on the screen should be multiplied to obtain its coordinates in the virtual physical plane [67].

Knowledge of the geometry of the road, geographical coordinates and the position of the point on the road image, allows us to obtain the physical coordinates of the object.

This can later be used to obtain GPS-like vehicle tracks (with longitude, latitude values) using only the data from the traffic camera.

Projective mapping is also known as perspective transformation (or homography transform), which is linear planar mapping.

Projective mapping can be achieved through the use of vector-matrix multiplication (formula 2). For an arbitrary object, the relationship of a fixed points in an image and a reference image can be described using forward-mapping matrix:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2)$$

There are a total of eight coefficients in the equations of the forward mapping for 2D-transformation (considering the eight degrees of freedom (two for each operation) – translation, rotation, scaling, shearing). Therefore, it is possible to map road representation on camera to any rectangle, that can be shifted, rotated, scaled and sheared in any way possible.

In this work, the camera calibration applied in determining the lane of the car: road scene projected on virtual rectangle to allow estimation of traffic parameters with lane precision. Example of virtual rectangle markup and result of transformation presented on figures 6 and 7.

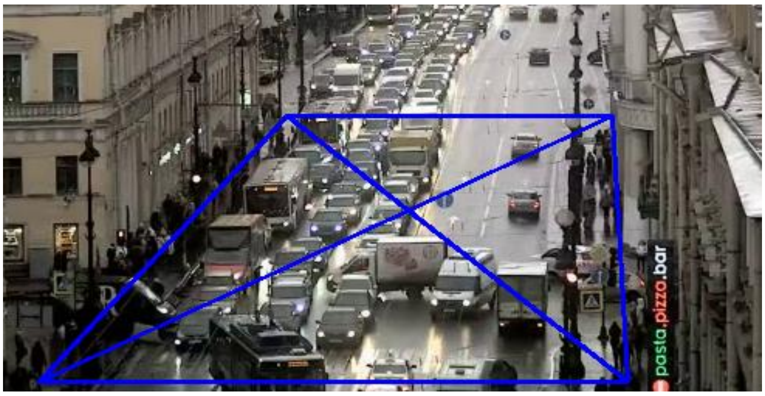

Figure 6 - Markup for perspective transformation

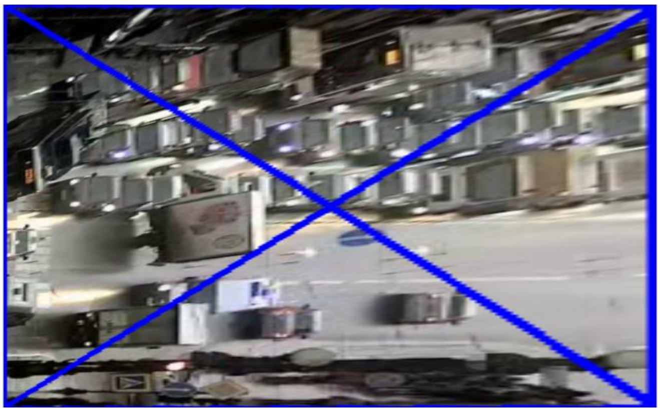

Figure 7 - Result of perspective transformation

2.13 Problem of traffic video analysis

Traffic congestion of observable road needs to be considered. Also, illumination varies dramatically over daytime. There are many things that can also affect quality of recognition.

Following are the possible issues with traffic cam usage:

1. Variety in types of vehicles (new, rare and unusual car models which can be absent in database used in detection methods training).
2. Work under wide variety of traffic conditions (congestion, rules violation, unusual behavior, car accidents).
3. Weather/light conditions (sun, twilight, night, night with road lights, falling or laying rain (wet road reflects light), falling or laying snow (can make background subtraction methods unfunctional, because of fast background change at snowfall), shadow strength and direction.
4. Computational efficiency of algorithm. Applicability for a real-time video-analysis.
5. Specifics of selected algorithm (e.g. color sensitivity/insensitivity, distortion (e.g. different scale of objects, rotation) sensitivity/insensitivity).
6. Physical errors (camera shake because of wind), video encoding errors (square noise).
7. Perspective-based distortion (depth of view, position and view angle of camera can make image of a car distorted and therefore affect quality of

recognition) and perspective-based effects (e.g. partial cars occlusion because of point of view).

8. Quality of video/cost estimation of camera.
9. Communication bandwidth (between datacenter and camera). Also significant for distributed video analysis.
10. Video quality. Video parameters, such as low frame rate can make tracking of already detected cars impossible. Video compression artifacts can add noise to a video. Low quality and CCD (charge-coupled device) camera noise also affect quality of recognition.

High traffic congestion, depending on point of view, can lead to a partial occlusion of cars in image, limiting applicability of background subtraction techniques.

Combination of different approaches to preprocessing should be used to achieve satisfactory results.

3 DEFINITION OF EXPERIMENT

3.1 An overview of experiment

The research is organized as follows:

1. Data collection (not less than a month of video).
2. Preliminary analysis – collected video is analyzed for optimal readout time period for estimation of traffic parameters, detected special behavior cases analyzed for average traffic flow and density.
3. Detection of special behavior cases.
4. Speed-density analysis – 7 days of video and hours related to special behavior are analyzed for speed and density of the traffic.
5. Modelling of special behavior: 1) choice and validation of classification model to distinguish normal road states (in space of speed-density) and related to special behavior 2) construction of simulation model to assess speedup from implementation of special behavior.

3.2 Data source

To collect data on cases of special behavior, a camera was selected with a view of the road section with dense two-way traffic on Nevsky Prospect, 110 meters long and 17.5 meters wide (Figure 8).

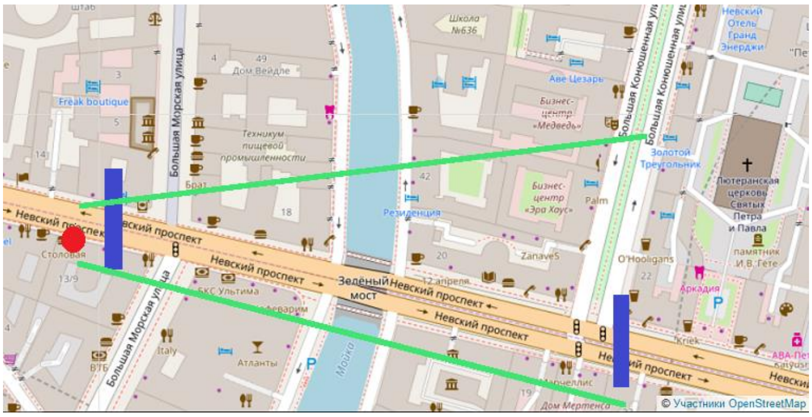

Figure 8 - Diagram of the camera overview

A set of video data contains video obtained for different periods of time:

1. 7 days in April 2017.
2. 14 days in November 2017.
3. 8 days in December 2017.
4. 6 days in January 2018.

A total of 35 days of video was collected and analyzed. The video has a resolution of 960 by 720 pixels. For video data, a mask of the region of interest was built to eliminate false positives of the object recognition algorithms in the image.

Initially direct data gathering was done with maximum video quality without recompression. But issue with high disk capacity requirements became explicit very soon(hard disk consumption is 780 MB per hour) – 67 GB of hard disk space consumed every 24 hours and method was reevaluated.

It was necessary to filter unnecessary parts of video using mask. In order to achieve mask, accumulated frame difference method[29] was used (figure 9).

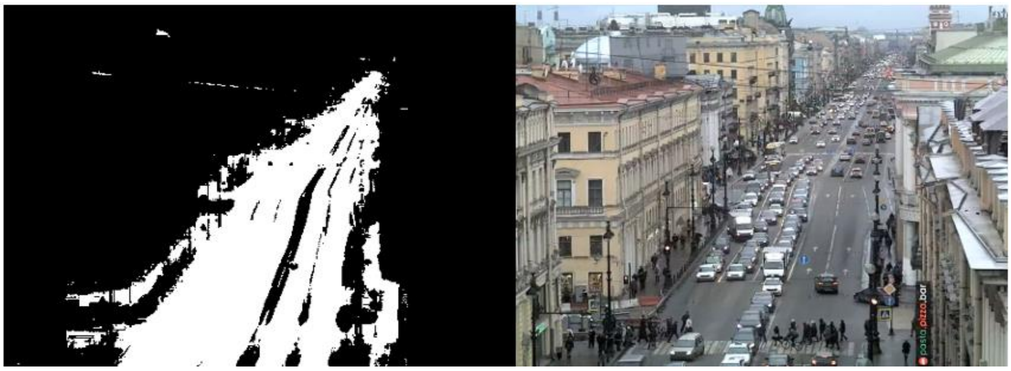

Figure 9 - Road mask generated using accumulated frame difference method(left) and initial image of the road(right)

As shown in figure 9, double line crossing (change of lane to opposite) occurs very rarely and therefore colored with black. This black line will not allow to continuously track vehicle that crosses double line. But it is necessary to detect such crossings. That is why the method was improved with double scaling and dilation. Also, not absolute frame difference was accumulated, but square of a difference to make facts of movements more significant in order to include rarely visited, but significant parts of the road.

3.3 Architecture of data collection system

The data collection system was implemented (Figure 10). The process of collecting and analyzing data consists of the following steps:

1. Collection of video files with duration one hour each.
2. Detection of cases of oncoming traffic - using computer vision methods, behavior that is characteristic to a special transport — movement by the oncoming lane is detected. At this stage, the entire array of video data is filtered for the presence of cases of special behavior in the video data.
3. Preliminary analysis of the traffic situation for traffic flow and traffic density for video fragments containing special behavior.

A unique feature of this system is the possibility of parallelism and pipelining of video data processing, which allows to build an optimized system for analyzing road traffic using computer vision methods.

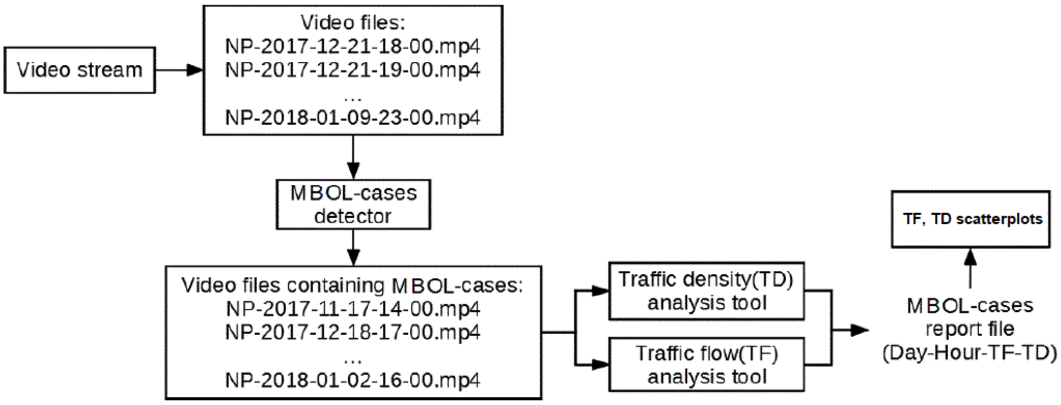

Figure 10 - Architecture of the data acquisition system (MBOL - movement by the opposite lane)

Based on traffic state data for the cases of special behavior, scatterplots are produced. Using plots, it is possible to suggest the degree of compliance of a traffic situation with the manifestation of special behavior in order to perform further analysis and investigation of video fragments for possible MBOL-cases.

3.4 Calculation of readout period

To determine the reading interval of traffic parameters, a second-precision analysis of the traffic density (using EDANet for semantic segmentation) was conducted within one hour at 11:00 am on the first day of video surveillance. Further, an analysis of the spectral density was made and the frequencies most characteristic of the traffic situation were identified. The graph (figure 16) shows that the frequency of 0.022 Hz is noticeable on all lanes. That corresponds to the period of 45 seconds (the mode of the traffic light). In accordance with the Nyquist – Shannon sampling theorem, it is necessary to take the Nyquist frequency as twice as high as the present frequency of 0.022 Hz, i.e. 0.044 Hz (period = 22.72 seconds). To simplify the calculations, a period of 20 seconds will be chosen (0.05 Hz).

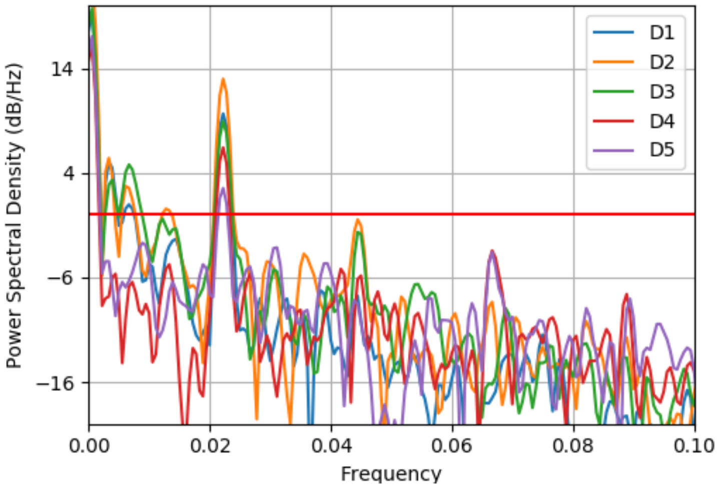

Figure 16 -Power Spectral density of traffic density (with second precision)

3.5 Preliminary analysis of optical flow

Dataset can be characterized with unstable frame rate (4-15) and video coding artifacts. This can produce unreliable optical flow measurements due to high pikes during delayed frames. Optical flow visualization produced noticeable flickering. Thus, preliminary analysis was conducted. Video scene has been analyzed for 175 frames with summation of total displacement magnitude for each frame. Unstable frame rate led to noticeable fluctuations in optical flow (Figure 17). Magnitude then was smoothed with moving average with 10-values window (orange curve).

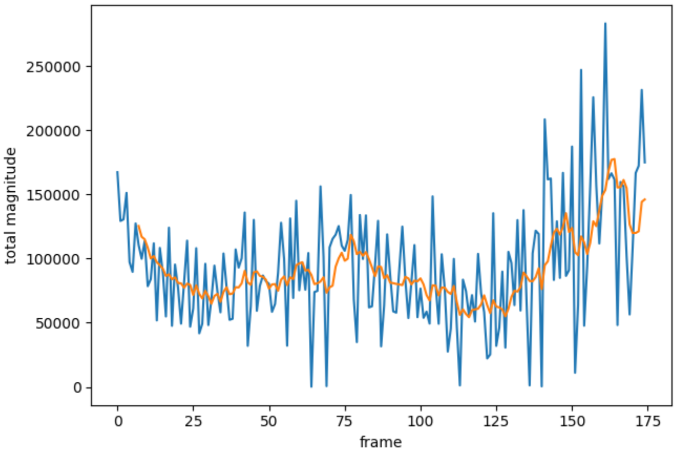

Figure 17 – Total displacement magnitude for every of 175 frames (using Gunnar-Farneback method)

3.6 Speed-density analysis pipeline

Traffic speed and traffic density parameters were selected to classify MBOL/non-MBOL-related 20 second time blocks (figure 11):

Frames from the video are obtained every 20 seconds (this value is calculated by analyzing the traffic density per second and identifying the maximum readout period of discrete readings in accordance with the Nyquist-Shannon theory - which will be discussed later).

Every 20 seconds (until they are completed), 8 frames are accumulated with an interval of 4 frames for analyzing the optical flow (by the Gunnar-Farneback method).

After calculating the optical flow, the optical flow map is filtered through the segmentation map. The frame processed by segmentation methods determines the presence of cars on the road. Thus, we obtain the speeds of only the necessary pixels (of the vehicles themselves). Density is also calculated for lanes (the number of “white” pixels in relation to the entire set of pixels is defined as the density of traffic on the road). Traffic speeds and densities are extracted separately for each lane.

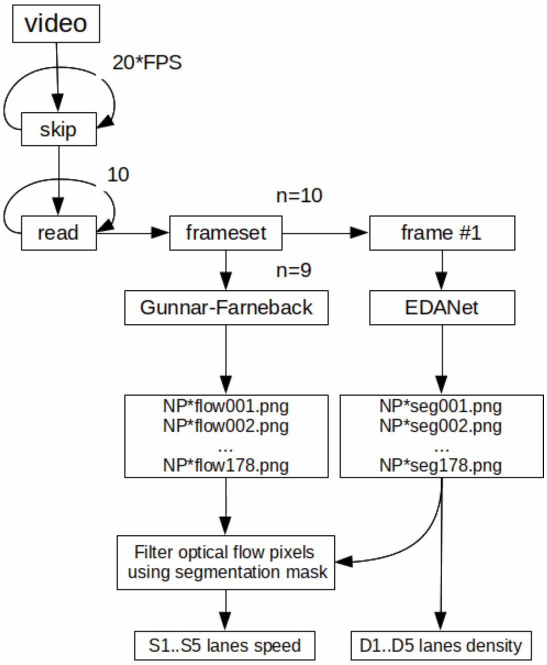

Figure 11 - Architecture of the speed-density processing pipeline

4 DETECTION OF SPECIAL TRANSPORT BEHAVIOR

4.1 Detection of special transport behavior using TrackLoops

As a special case of special behavior, a movement by the oncoming lane (MBOL) was chosen.

To detect the movement in the oncoming lane, a novel method named “TrackLoops” is used (Figure 3), which uses tracking windows located in certain places of the road (their location and shape can be optimized). Each tracking rectangle is placed on the road area for a limited number of frames.

At each location, an object located in a rectangle is captured and tracked over several subsequent frames. Then - reset and return the rectangle to its original state with record of displacement (dx and dy along axes x and y). If the rectangle moves along the direction of the road, then no special behavior is detected. But if it moves against the direction of the road several times in a row, then the detector is triggered, gives a signal, saves a screenshot and records the event in the log.

For tracking rectangles, the Kernelized Correlation Filters (KCF) method[68] was used, which allows tracking objects with high performance (necessary for Big Data) and fixed tracking window size.

Further, according to the journal, the list of video files containing movement by the oncoming lane is restored.

This method uses a periodic reset of the detector, since computer vision algorithms designed to track objects are subject to track loss error [8]. Resetting the tracker allows one to restore tracking in a short period of time. Thus, the method of detecting special behavior, as compared with the methods of searching and tracking special transport, is more resistant to errors.

This method is similar to using Virtual Loop to count machines per video, but it uses a tracking algorithm, instead of the often-used combination of background subtraction and baselines.

Also, cases of false triggering of the detector were detected. These were cases of movement of the reflection of headlights on wet asphalt in the oncoming lane. However, since the headlights extend to a limited area of the road, it was possible to eliminate this error for several cases, by finding the optimal placement of the rectangles on the road (with the placement at a greater distance from the double line).

The TrackLoop method is the placement of the tracking window at specific locations in the scene for a limited period of time. The method does not use a detector and therefore is not subject to errors of the first and second kind in the problems of detection. Also, the method is more resistant to the “track loss” error, since every few frames (for example, 4) the tracker is reset and initialized. An example of placement of tracking windows is shown in Figure 12.

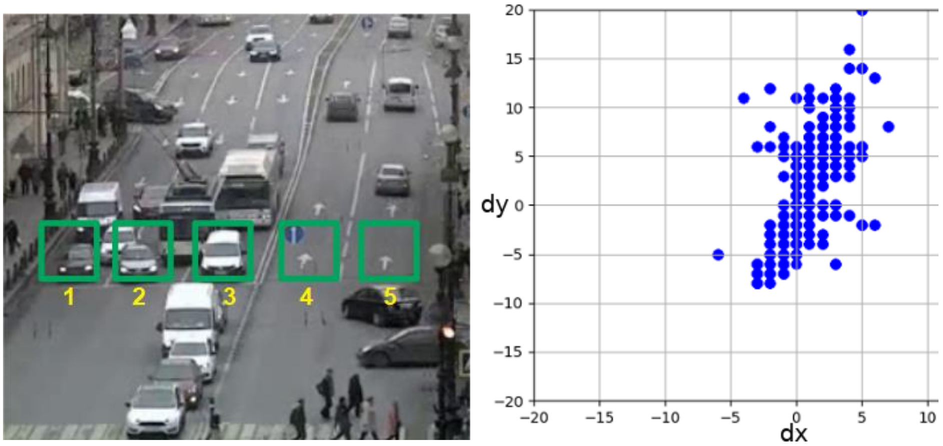

Figure 12 - An example of tracking windows “TrackLoop” located on the road and map of displacements for the second “TrackLoop” on the right

5 ANALYSIS OF SPECIAL TRANSPORT BEHAVIOR

5.1 Traffic flow estimation

Traffic flow estimation is implemented using the object detection and tracking approach [9] in a limited area of the image. This method is a cyclic procedure involving the use of the Viola-Jones method to search for objects in an image and MedianFlow for subsequent tracking of detected objects. The intersection of the tracked object by the center of the rectangle means an increase in the traffic flow counter for the corresponding side of the road. The side of the road on which the vehicle is moving is determined using the projective transformation method. The use of this method is due to the frequent overlap of machines with dense traffic, at a certain camera viewing angle, which prevents the use of background subtraction and segmentation, as well as Virtual Loop, to calculate the traffic flow traffic, as they are used to estimate traffic parameters on highways, where cars move at a distance.

To implement vehicle counting, vehicle detection and tracking methods were combined into well-known detect-and-track approach[23,66,69,70]. In the implementation of vehicle tracking it was observed that tracking window can “fall” from a vehicle on a road and lay there with zero speed. In order to filter such windows, “time-to-delete” timer was introduced for all the tracking windows (it allows to delete “fallen” windows without any commitment into results). “Exit” rectangles are used to delete successfully traveled cars (figure 13). After deletion, track data calculated and written to the report file.

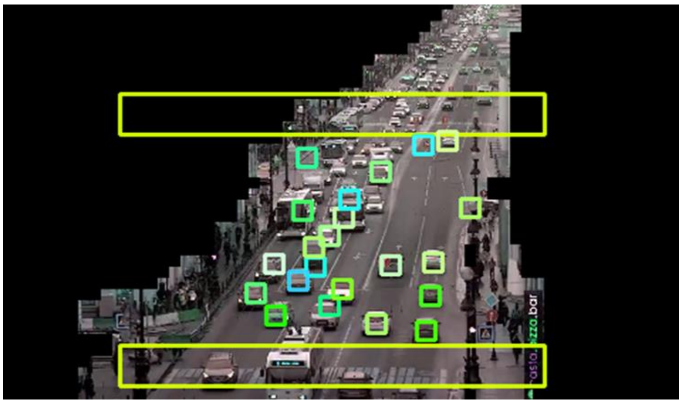

Figure 13 - vehicle tracking and counting using “Exit” rectangles

5.2 Implementation of vehicle detection methods

In the implementation of vehicle recognition, Viola-Jones method was used. The car recognition program was implemented using ready to use cascade (Figure 14) based on UIUC Image Database for Car Detection(collected by Shivani Agarwal, Aatif Awan and Dan Roth), which based on 1050 training images (550 car and 500 non-car images).

Problem of insufficient recognition quality lies in point of view used in positives dataset – vehicles are photographed from the side, but on a test video, cars observed from the top of the building on the side of the road at an angle to the horizon. Thus, it is necessary to gather the most suitable positive and negative images dataset in order to train new cascade.

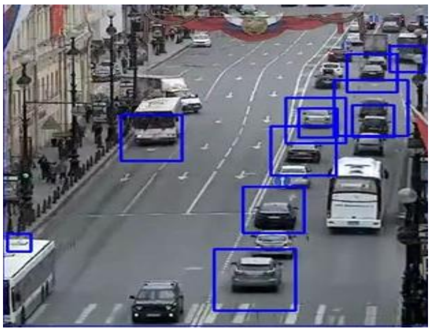

Figure 14 - Quality of recognition of ready to use Haar cascade

For the case, vehicle counting implementation using baseline or virtual loop is not appropriate due to vehicle occlusion.

Vehicle detection using HOG descriptors with SVM classifier was also implemented (Figure 14).

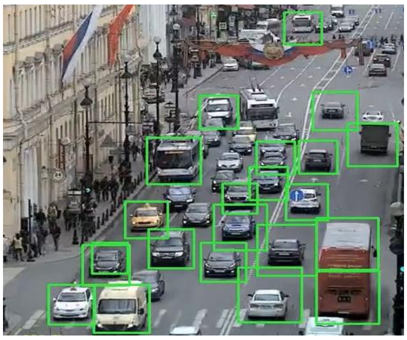

Figure 14 - recognition using HOG+SVM method

Also, a method of detecting cars based on background subtraction and blob tracking was implemented (figure 15). For the implementation of background subtraction method, BGSLibrary was used. The library contains implementation of multiple types of background subtraction methods implementations (e.g. gaussian mixture model(GMM)).

Occlusion of blobs is happening because of shadow and partial occlusion of cars in initial image. Background subtraction can be used for vehicle detection only if vehicles maintain distance.

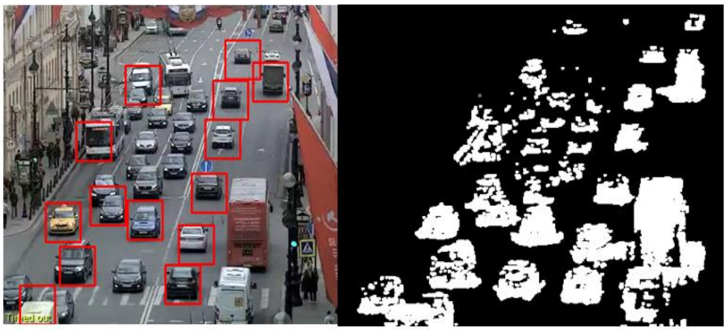

Figure 15 - Recognition using background subtraction (left) and result of background subtraction with the use of morphological operations (right)

Many vehicles are not detected as blobs because of small difference in road and car windshield color (and also color of vehicle surface reflection). It produces many small unconnected blobs of which the vehicle consists.

6 RESULTS

A total of 35 days of video data from a road surveillance camera were collected and analyzed.

When using the TrackLoop detector, 27 cases of exit to the oncoming lane were found (figure 16). In 25 cases it was a special transport, in 2 cases it was a violation of the rules of the road by an ordinary vehicle.

For video clips lasting one hour, a density and flow analysis were performed using the road occupancy estimation through edge counting and detect-and-track approach (using Viola-Jones and MedianFlow methods) correspondingly. The method based on the detection of special behavior, showed high resistance to light interference. This type of interference is most typical for evening hours when there is a high traffic load associated with the return of people from work and most likely to travel to the oncoming lane due to the inability to quickly cover the distance along the main lane.

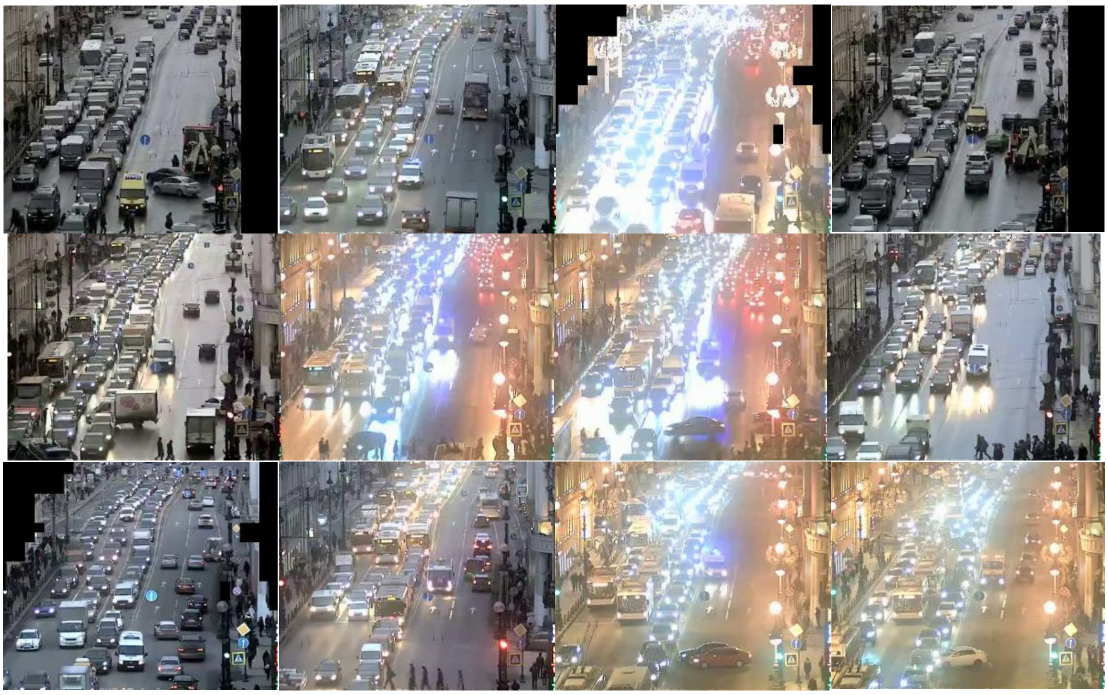

Figure 18 - Examples of cases of special vehicles on the oncoming lane

6.1 Comparison of vehicle recognition methods with manual markup.

2284 positive and 5171 negative images (of vehicles and background objects correspondingly) were collected into big training dataset, 800 positives and 2700 negatives into small training dataset. Haar cascade has been trained up to 17 stages.

Implementations of all of the methods were modified in order to conform with performance and quality of recognition test process.

Ten frames of test video were selected for markup comparison. Manual markup was done. Results presented in figure 2. Haar and HOG related to small training dataset and Haar2+and HOG2+ related to big training dataset. BGS does not depends on dataset. When there is high traffic density, car partial occlusion occurs that reduces BGS quality of recognition significantly. BGS also has more false positives than other methods.

It can be concluded, that BGS and HOG are both outperformed by HAAR in terms of speed (figure 19) and quality of recognition (figure 20). Thus, Viola-Jones method will be used as a visual object detection method in detect-and-track approach for traffic flow estimation.

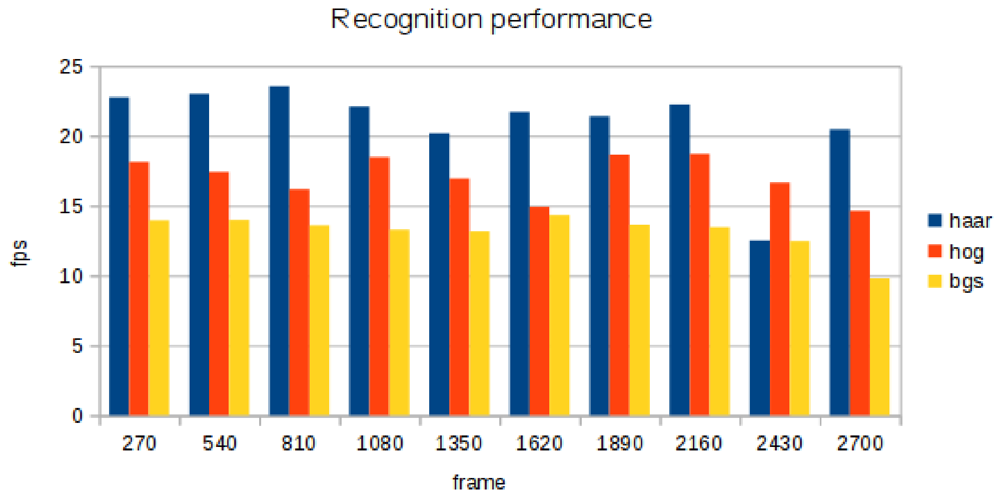

Figure 19 - Recognition performance comparison using HOG+SVM, Haar cascades, background subtraction methods

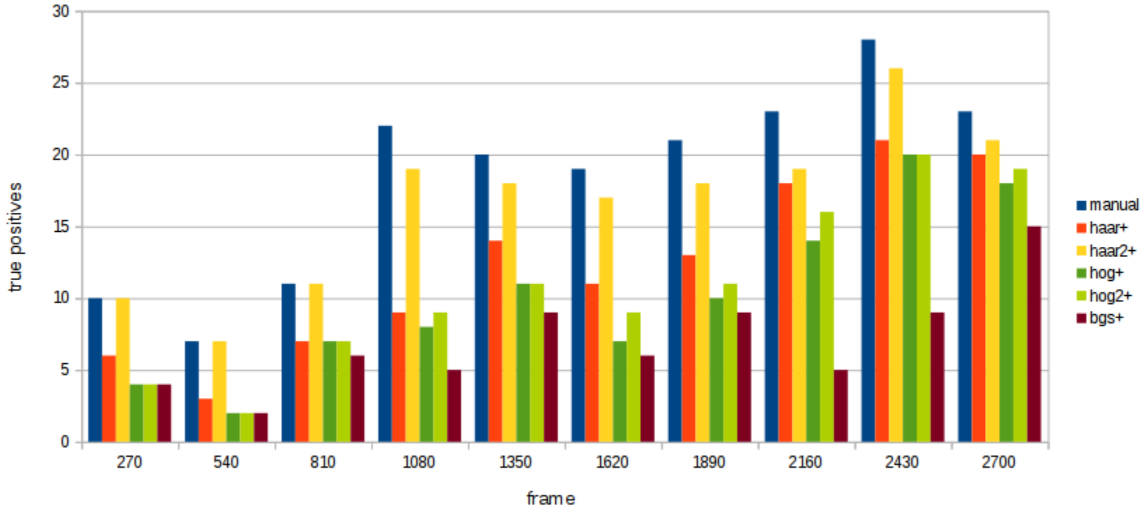

Figure 20 - Quality of recognition comparison using HOG+SVM, Haar cascades, background subtraction methods

6.1 Application of real-time inference neural network architectures

Neural network architectures for semantic segmentation with ability of real-time inference were assessed for applicability to the tasks of analyzing the density and count of traffic

60 frames were assigned for the task of determining the amount of traffic and 60 frames for the task of determining the traffic density. The samples were divided in the ratio of 50 to 10 (for training and validation). For each sample, augmentation was performed (12 different transformations), including variations of scaling, blurring, shift.

EDANet and BiSeNet architectures were used as real-time neural output networks; SegNet neural network was used as the baseline (Table 1). Processing speed is compared on images of 512 by 512 pixels.

Table 1. Comparison of neural network architectures on the figure segmentation task

Architecture	Parameters, mln	Epochs	Dice train loss	Dice val loss	Frames per second
BiSeNet	0.88	40	0.05	0.14	95
EDANet	0.68	8	0.041	0.09	64
UNet	13.4	3	0.037	0.097	23

EDA significantly outperforms BiSeNet in terms of learning speed (a smaller loss was reached already at the 8th epoch) with a smaller number of parameters, but the output speed is about one and a half times less. U-Net very quickly captures the details of the scene due to skip connections, but it requires more memory due to the symmetry of the architecture. Despite the high speed, BiSeNet has a lower learning rate and worse accuracy. For the segmentation task (for estimating traffic density), EDANet seems to be the most efficient. Visualizations of the work of the EDANet, BiSeNet and UNet methods are presented in Figures 21-24.

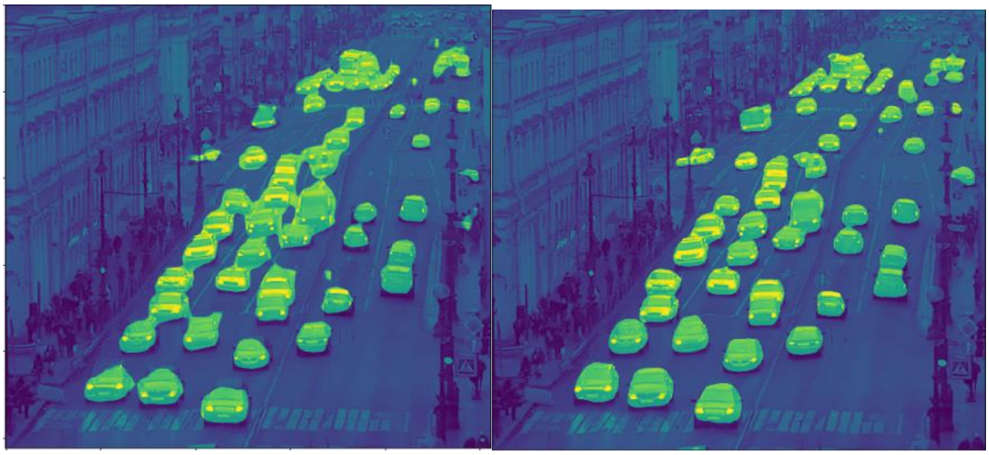

Figure 21 - Result of work on a randomly selected frame of BiSeNet architecture after 40 epochs (left) and U-Net after 3 learning epochs (right)

In the case of point segmentation, the number of pixels of different classes on the markup maps is unbalanced and the value of the Dice metric does not indicate the nature of the error in approximating points, therefore visualization is necessary. Since asymmetric networks rely on scaling in the last layer, there is a specific limit of granularity for them. This limit can be reached during training (table 2). The point segmentation task requires small details as a result of the output (points). Therefore, the use of asymmetric architectures that perform scaling in the last layer does not seem appropriate for point segmentation and in practice does not produce an acceptable result.

Table 2. Comparison of neural network architectures on the point segmentation task

Architecture	Parameters, mln	Epochs	Dice train loss	Dice val loss
BiSeNet	0.88	40	0.32	0.72
EDANet	0.68	10	0.31	0.64
EDANet	0.68	20	0.24	0.64
EDANet	0.68	30	0.22	0.63
EDANet	0.68	40	0.22	0.63
UNet	13.4	10	0.11	0.61

When using asymmetric architectures, there are scaling artifacts and generally low accuracy compared to symmetric Unet architecture.

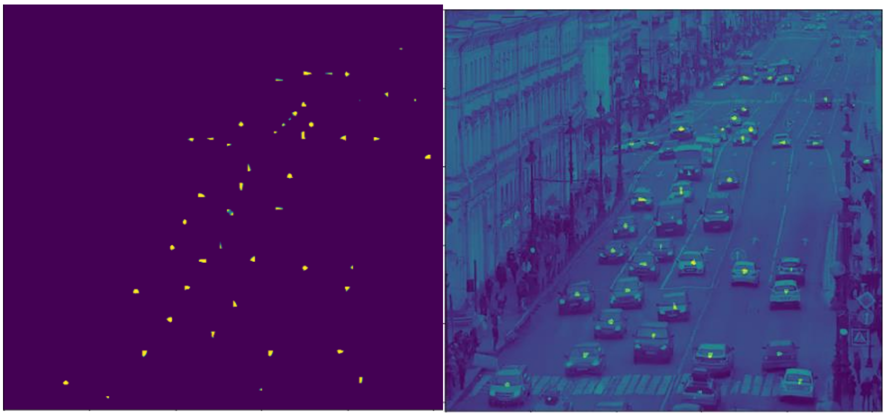

Figure 22 - Result of point segmentation on a randomly selected frame using EDANet architecture (20 epochs)

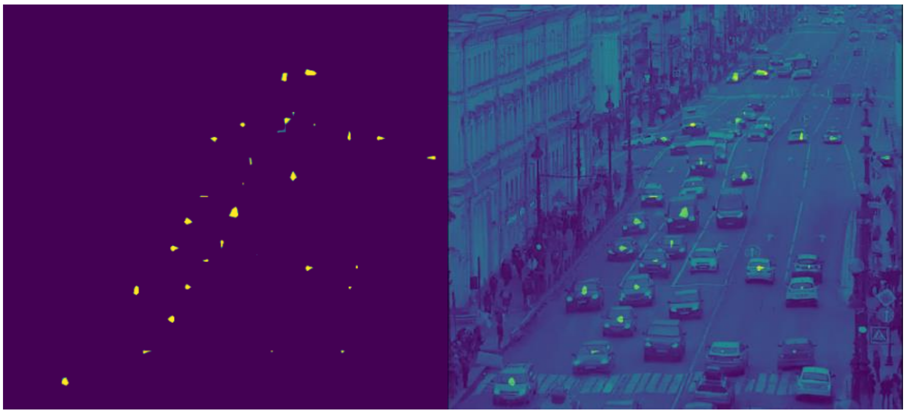

Figure 23 - The result of point segmentation on a randomly selected frame using the BiSeNet architecture (40 epochs)

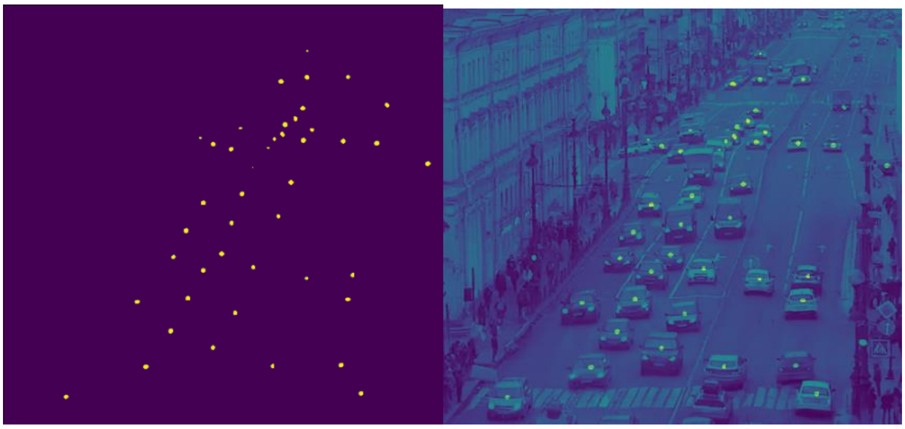

Figure 24 - The result of point segmentation on a randomly selected frame using the UNet architecture (10 epochs)

According to the results of point segmentation, it can be concluded that to solve such a precise task as a point segmentation, a symmetric architecture is needed (in which there will be no scaling distortion). Therefore, it is recommended to use the UNet architecture for point segmentation, and to compensate for the impossibility of real-time data processing by frame skipping.

Despite the presence of a large number of extraneous objects, only cars are asegmented during figure segmentation (Figure 25), which indicates a significant difference of this method from the Background subtraction [8].

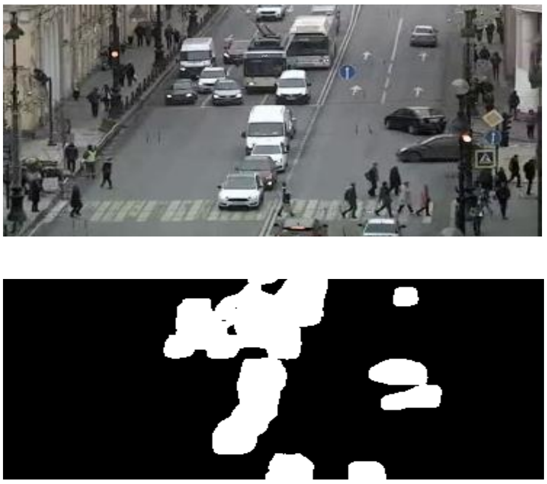

Figure 25 - An example of the selectivity of the segmentation method

An example of counting the number of cars for one hour in which congestion is observed is shown in Figure 26. Presumably, this method can also be used to estimate the density of pedestrian traffic and count the number of pedestrians.

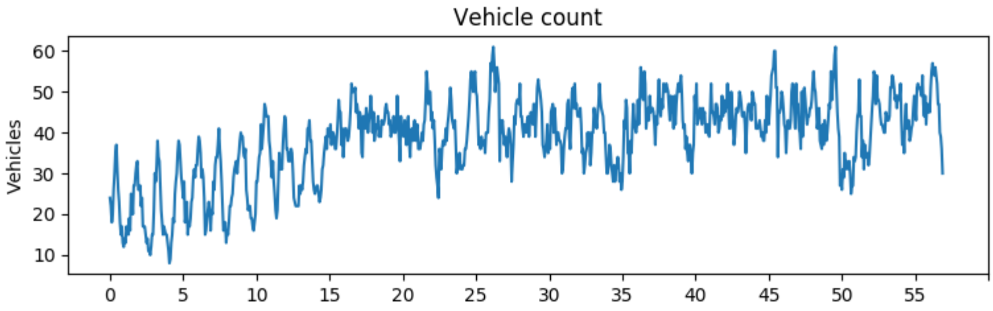

Figure 26 - Graph of the number of cars. Measurement time from 14:00 to 15:00, November 17, 2017

By the nature of the change in the number of cars, it is possible to determine the state of the traffic flow (traffic phase): in the absence of a traffic jam, cycling with a high amplitude is observed; The graph shows that by the middle of the hour a traffic jam is formed with heavy traffic.

Also, there is a graph of the number of cars for the first 5 minutes of the hour (Figure 27). Measuring the distance between the “waves” of traffic, one can evaluate the mode of operation of the traffic light.

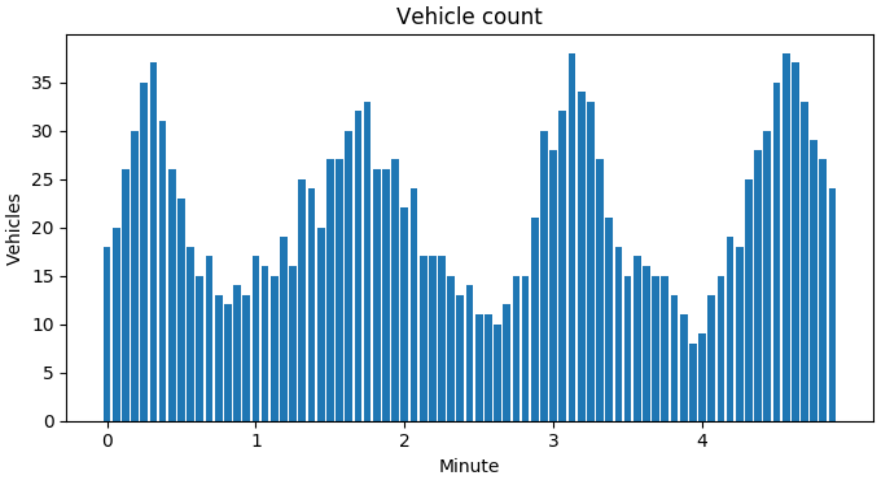

Figure 27 - Graph of the number of cars in the 5-minute interval

In the future, this method will become a component of the system for analyzing traffic parameters in the system for detecting cases of special behavior. Traffic analysis using video surveillance tools provides an opportunity to estimate the exact number of cars on the road with an accuracy of a second and with an accuracy of a lane. This feature can be used to build micromodels of traffic on the observed section of the road. Many modern traffic analysis services, including Yandex.Probki [71] and Google Traffic [72], cannot provide such accurate data. they rely on GPS data having an error of 5 meters in an open area [16].

6.2 Traffic density for 7 days of observations

Using semantic segmentation, 20-second blocks during 7 days of observations, traffic density with lane precision was analyzed (figures 28-29). Left (oncoming) lanes of the road can be characterized as the densest (related to high traffic flow from the city center). Right lanes are less dense and have unused capacity. Considering average and maximum densities, two certain roads can be replaced with one road. The densest lane is D2 which seems to be the lane of choice for vehicle drivers. D1 and D3 in usage are secondary in relation to D2.

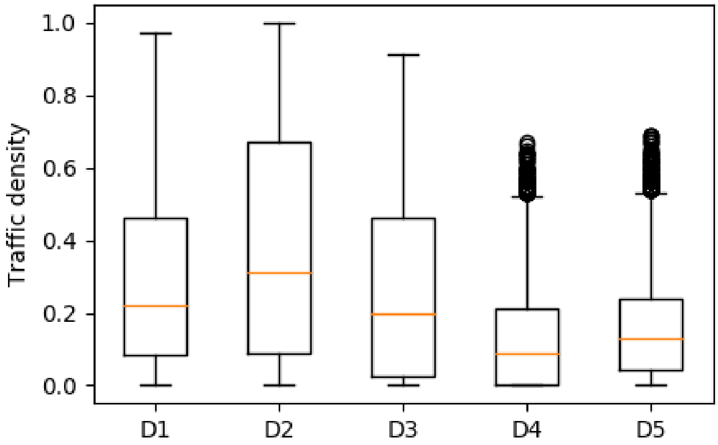

Figure 28 – Boxplot for traffic density of time blocks when MBOL did not occurred (with lane precision)

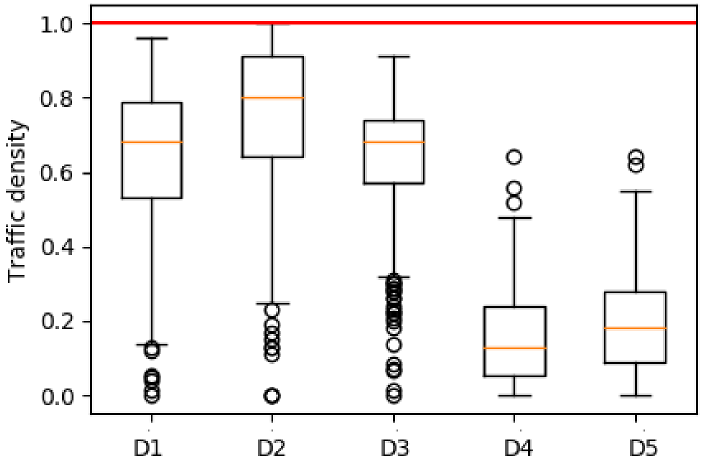

Figure 29 – Boxplot for traffic density for time blocks when MBOL occurred

6.3 Traffic speed for 7 days of observations

Using semantic segmentation and Gunnar-Farneback optical flow calculation, 20-second blocks during 7 days of observations, traffic speed with lane precision was analyzed (figures 30-31). S1,S2 and S3 are characterized with slow traffic movement with very rare observations close to speed limit (plot shows that vehicle drivers follow speed limit in the city (60 km/h)). S4 and S5 have almost twice higher speed values than S1-S3. All of the lanes have speed limit exceeding except S3 (but this probably related to insufficient number of observations). For congested road state, speed is lower for lanes

1-3, but for lanes 4-5 it is a bit higher. Possible explanation is specific transport state, which related to high outflow of traffic from city center during evening. It can also explain higher speeds on lanes 4-5 due to lower traffic flow (it means less density) to city center at the evening.

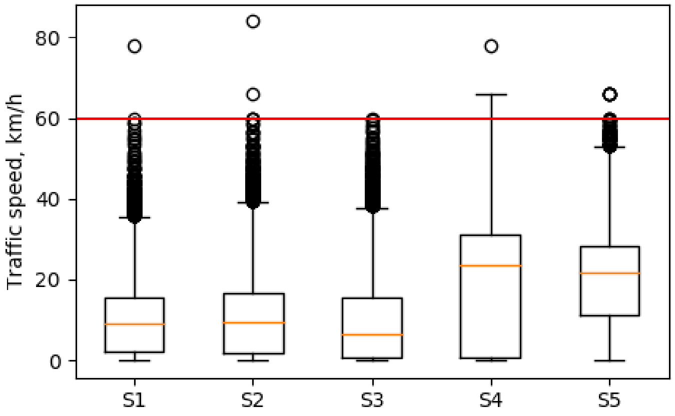

Figure 30 – Boxplot for traffic speed of time blocks when MBOL did not occurred (with lane precision)

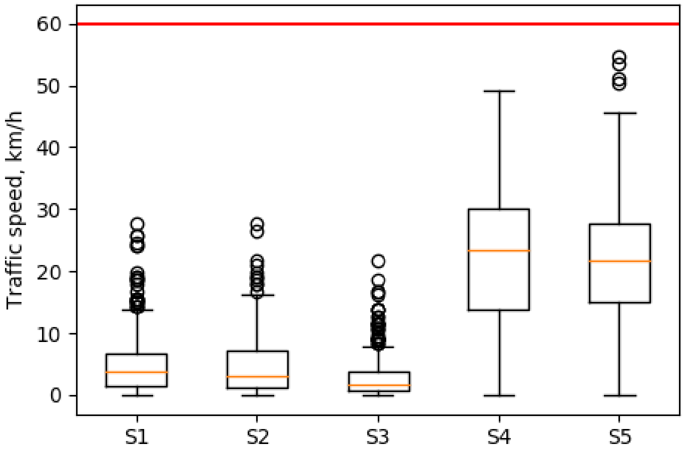

Figure 31 – Boxplot for traffic density for time blocks when MBOL occurred

6.4 Adaptive suppression method

New method named “Adaptive suppression” has been developed to detect anomalies in data set.

Adaptive Suppression - a method of detecting data anomalies, in which all data points have an initial accumulator value that interact at a distance and reduce each other's accumulator values.

The procedure of the method is as follows:

1. All data elements have an initial value of accumulator 1.0 and are gradually added to the value map.
2. Adding a new value if there is another value nearby causes a mutual suppression operation and a decrease in the accumulator value of both cells by a certain amount, called the mutual suppression coefficient, which is associated with the current accumulator value.
3. The more “suppressed” the neighboring point, the more it suppresses the new one that is next to it. Thus, the storage of environmental information in the composition of each cell is provided. Thus, a dense set of points is distinguished by a strong mutual suppression, while the far and rarely located points retain their accumulator value.

Accumulator value totals display a measure of the anomaly of each point in the data. The similarity of data points is estimated based on the distance metric (measure of difference). Anomalies can be calculated for data for which there are distance metrics exist.

The method can be implemented in many ways:

1. Single suppression. The coefficients are selected to ensure the output of 95% of points beyond the limit of 0.1 (at the level of the battery).
    a. Suppression can be implemented by successive points. In this case, the traversal order affects the nature of the suppression.
    b. Suppression can be realized by remembering the influence, without changing the values of the batteries during the bypass process. Upon completion of the crawl all effects are applied. This allows you to

remove the influence of the order of bypass points. It turns out that the values of the batteries with one saving bypass are not considered.

2. Repeated suppression. Odds are selected several times. Providing the weft of a defined share of points beyond the limit of 0.1 at each of the selection steps.
    a. It can be implemented through a sequential change in the values of accumulators on each pass, with regular randomization of the order of bypassing points.
    b. Or, by using the preservation and application of effects at the end of each iteration.
3. Each of these options can be implemented using a decreasing influence with distance (using kernel), or by using a static interaction radius (a point affects only points in a given small radius). The use of a static radius (integer) does not reduce the flexibility due to the possibility of twisting the real interaction coefficients.

Initially, all data points represented as accumulator cells have initial accumulator value (e.g. 1.0). After iteration of cross-suppression (depending on the distance), accumulator values decreased. Example of histogram of accumulator values presented on figure 32. Method receives a hypothesis about percent of anomalous data points in dataset (e.g. 5%). Considering anomality percent, we can further perform iterations of suppression, until 95% of accumulator values will be lower than specified threshold (e.g. 0.1). Thus, method named adaptive because of necessity of iterative suppression until percent of data points marked as normal will be reached and anomalous data points will last above anomality threshold. Method produces not only mark of anomality, but also value of anomality (from 1.0) represented as retained accumulator value.

A unique feature of the anomalies, when applying this method, is the fact that particularly anomalous points retain their accumulator value for a long time while other points are suppressed. Therefore, the outflow rate of the accumulator value can also be considered in the determination of anomaly. The more “suppressed” the neighboring point, the more it suppresses the new one that is next to it.

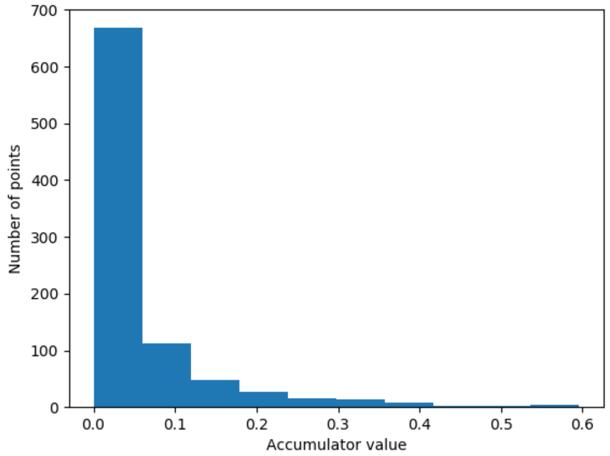

Figure 32 – Example of histogram of accumulator values after cross-suppression

Given that the method operates only with a distance metric (a measure of similarity), we can conclude that the method is applicable to the search for anomalies in datasets of images, sounds, documents, user comments, user profiles in social networks, any type of data for which similarity metrics exists. There is a method for obtaining a universal representation of texts [73]. Thus, having a similarity metric for such representations, one possibly can search for unique (anomalous) texts or even perform a plagiarism check (uniqueness / anomality).

6.5 Using the TrackLoop / TrackGrid and Adaptive Suppression methods to detect anomalies in traffic dynamics

Dataset QMUL Junction[74] is a dataset containing 1 hour of video divided into time sections (a total of 1800), of which 45 contain abnormal situations (prohibited U-turns, driving against the direction of traffic movement, stopping traffic). The dynamics of the scene based on the value of the TrackGrid elements (which are uniformly distributed TrackLoops) are shown in Figure 33. Anomalous time segments are not visually distinguished when performing dimensionality reduction (figure 34).

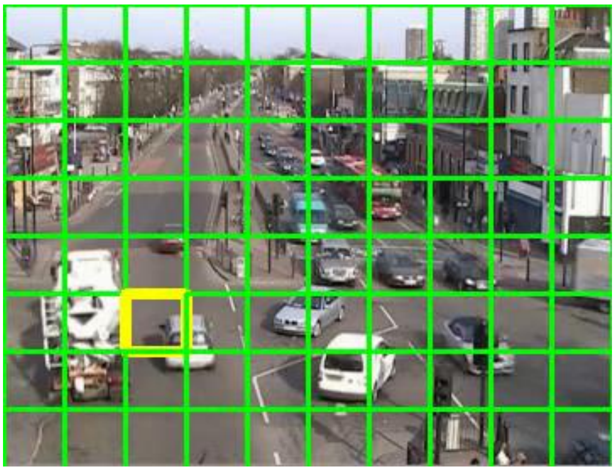

Figure 33 - Application of the TrackGrid method to the QMUL Junction dataset

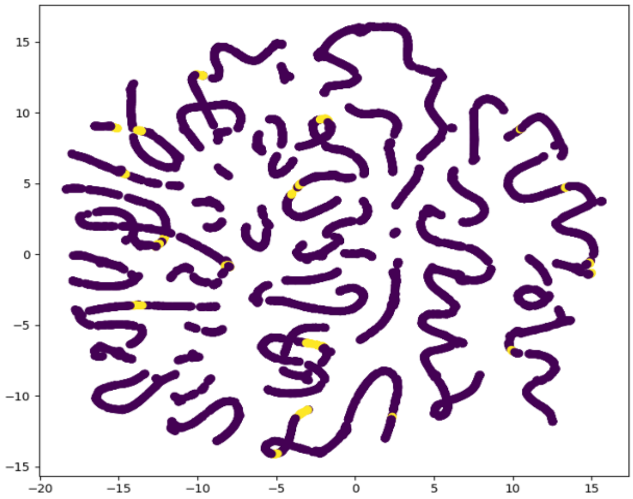

Figure 34 - Applying the UMAP dimension reduction method to the TrackGrid results for the QMUL Junction dataset

Since the method reads the dynamics of the scene, not the visual representation, anomalies of the dynamics of the scene are highlighted. This leads to the fact that the situations of movement of objects of anomalous shape, such as double-decker buses, detected as anomalous (Figure 35). Also, prohibited U-turns are detected.

Using the method, it was possible to achieve AUC = 0.726 using the adaptive mutual suppression method based on the values of the TrackGrid elements and the “average” aggregation function (Figure 36). Aggregation by quantile 0.8 increased the AUC value to 0.746. The result obtained is comparable to other studies conducted on this dataset (Table 3).

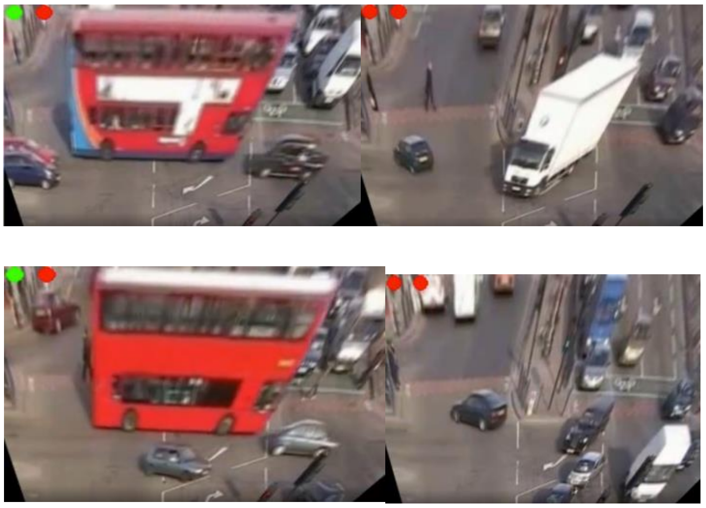

Figure 35 - Anomalies identified using TrackGrid and Adaptive Suppression methods

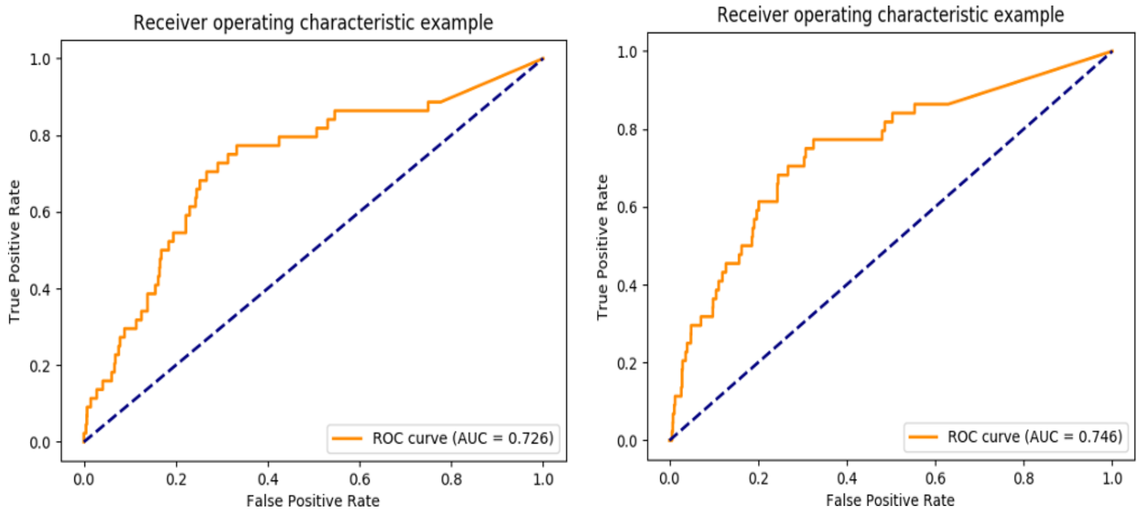

Figure 36 - Graph of the ROC-curve for the function of aggregation of accumulator values using aggregation by “average” (left) and “quantile 80%” (right)

Also it suggested using the False positives over true anomalies curve (FPTA, Figure 37). Vertical 5.0 corresponds to an acceptable ratio of the number of false positives of the detector and the true number of anomalies (it depends on the task and is set by the experimenter). Anything beyond the limit of acceptability (red vertical) does not interest us, since the result, with this ratio of false and correct detections, does not suit us. The green dotted line marks the line of an acceptable detector that ensures compliance with the acceptable ratio of false and true alarms at different scales.

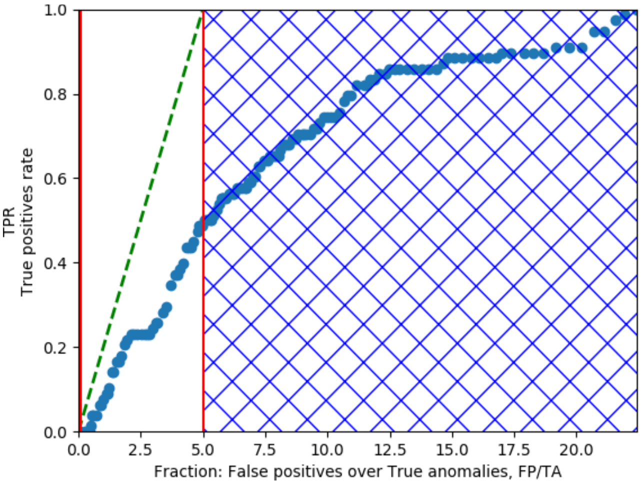

Figure 37 - The curve "False positives over true anomalies"

Table 3. Comparison of the result

Name of the article	Year	AUC
Boiman et al. Sparse inference by composition	2007	0.618
Roshtkhari et al. Online Dominant and Anomalous Behavior Detection in Videos (Dense spatio-temporal patches)	2013	0.687
Loy et al. Stream-based Active Unusual Event Detection	2010	0.7023
Isupova O. Anomaly Detection in Videowith Bayesian Nonparametrics	2016	0.71
Proposed method	2019	**0.746**
Loy et al. Modelling Multi-object Activity by Gaussian Processes	2009	0.7643
Vagia K. et al. Multiple Hierarchical Dirichlet Processes for Anomaly Detection in Traffic	2017	0.8657
Del Giorno et al. A Discriminative Framework for Anomaly Detection in Large Videos	2016	0.895
Loy et al. From Local Temporal Correlation to Global Anomaly Detection	**2008**	**0.9765**

6.6 Traffic density and traffic flow for MBOL-hours

For video clips lasting one hour each, containing instances of special behavior, a density and traffic flow analysis was performed. Scatter plots were constructed for traffic density for the left and right sides of the road (Figure 38). There is a noticeable positive linear correlation (R = 0.724) for the values of traffic density on the main (left) and opposite (right) sides of the road.

Departure to the oncoming lane occurs at high traffic density values on the left (main) side of the road (more than 45% black pixels), with low traffic density on the right (oncoming) side of the road (from 15% to 35% black pixels). When determining the traffic density, a ratio of the number of black and white pixels (after application of Sobel operator to the image) is calculated.

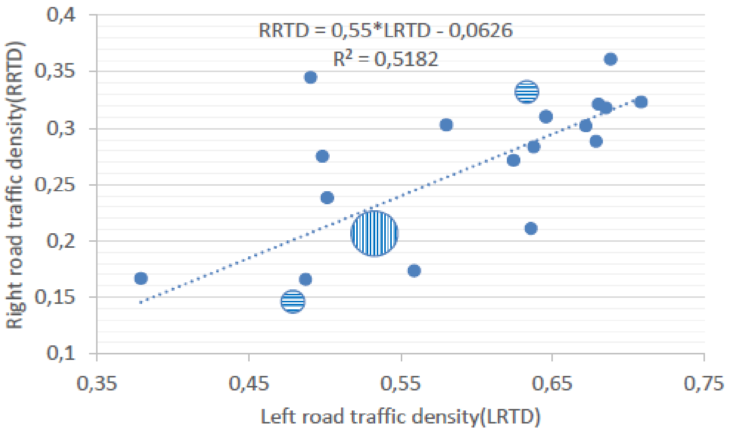

Figure 38 - Scatterplot for traffic density on the left and right side of the road

Scatter plots were constructed for the traffic flow for the left and right sides of the road (Figure 39). There is a noticeable positive linear correlation (R = 0.64) for the flow values on the main (left) and opposite (right) sides of the road. Departures to the oncoming lane occur at high traffic flow (more than 800 cars per hour) on the main side of the road, with significant variations in the flow for the opposite side of the road (from 550 to 1350 cars per hour).

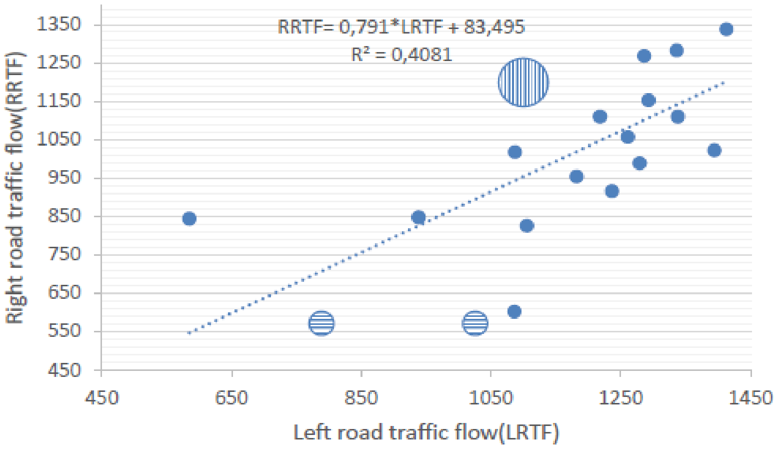

Figure 39 - Scatterplot for traffic density on the left and right side of the road

7 MODELING OF THE SPECIAL BEHAVIOR

7.1 Decision model based on a traffic density

Video has been analyzed with 20 seconds granularity. Normal 20-seconds blocks from 7 days of observations (with lane precision) were analyzed for density and speed and were classified using KNN classifier against MBOL-blocks. Totally, 30476 observations: 29795 negatives and 681 positives related to 20-seconds blocks when MBOL occurred with +5/-5 minutes interval covering.

Road images has been segmented (using EDANet) for vehicle presence and projectively transformed. Areas related to each lane (D1-D5) were extracted. Speed has been extracted accordingly, by using 10 frames accumulation of Gunnar-Farneback method results with extraction of meaningful areas using obtained segmentation image as mask.

By using TPOT Python Automated Machine Learning Tool, using 5-Fold Cross-Validation and F1-score as a target metric, best fit model was discovered: it is KNN with 4 neighbours with Euclidean distance as a metric.

Evaluation of KNN with 5-Fold Stratified Cross-Validation gives average F1-Score 0.88. As can be seen from confusion matrices, classifier detects actual (Act*) MBOL-cases finding near 90% of MBOL-related blocks correctly.

Second best classifier is Decision Tree (DT). DT has average F1-Score 0.39 for 6-level depth (feature importances presented in figure 40) and 0.55 for 12-level depth. Significant difference in F1-Scores of KNN and Decision Tree (0.88 and 0.39 correspondingly) notes the specifics of the data, for which estimation of nearest MBOL-related data point is more effective than making complex decision tree.

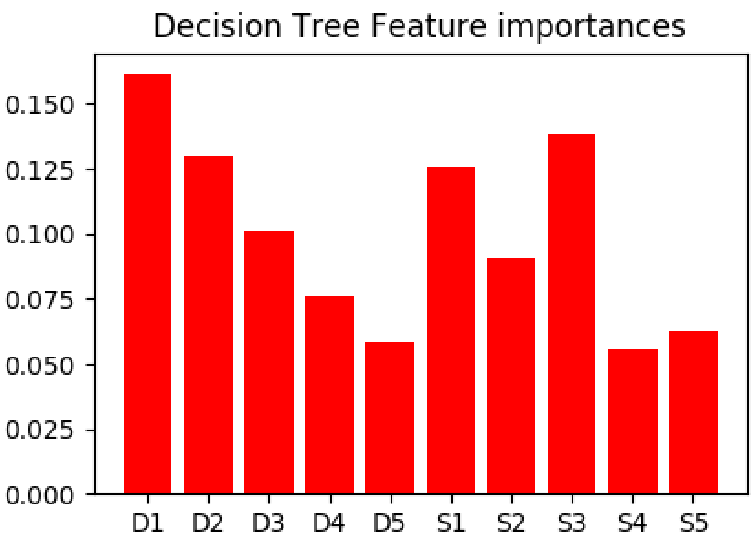

Figure 40 – Decision Tree feature importnaces (depth=6)

Decision Tree for depth=6 has been constructed (figures 41-42). Density threshold (0.522, 0.599) on the first two lanes has determining influence on decision of MBOL-case. Low speed on the 1st lane (less than 5.6 km/h) or 3rd lane (less than 4 km/h) also can characterize the road situation as congested. Absence of vehicles on 5th lane with low speed on third lane notes possibility of MBOL.

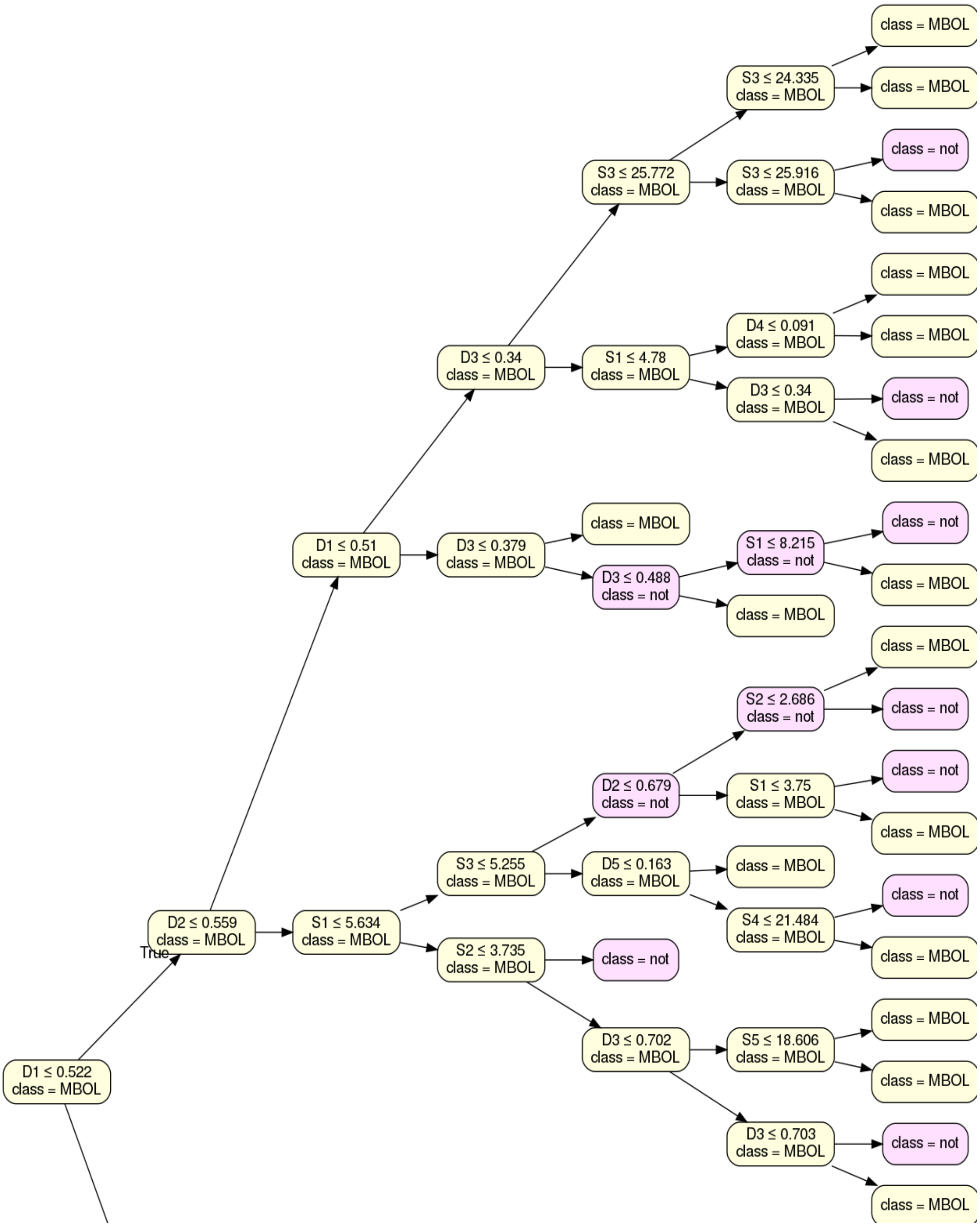

Figure 41 – Decision tree model (part1)

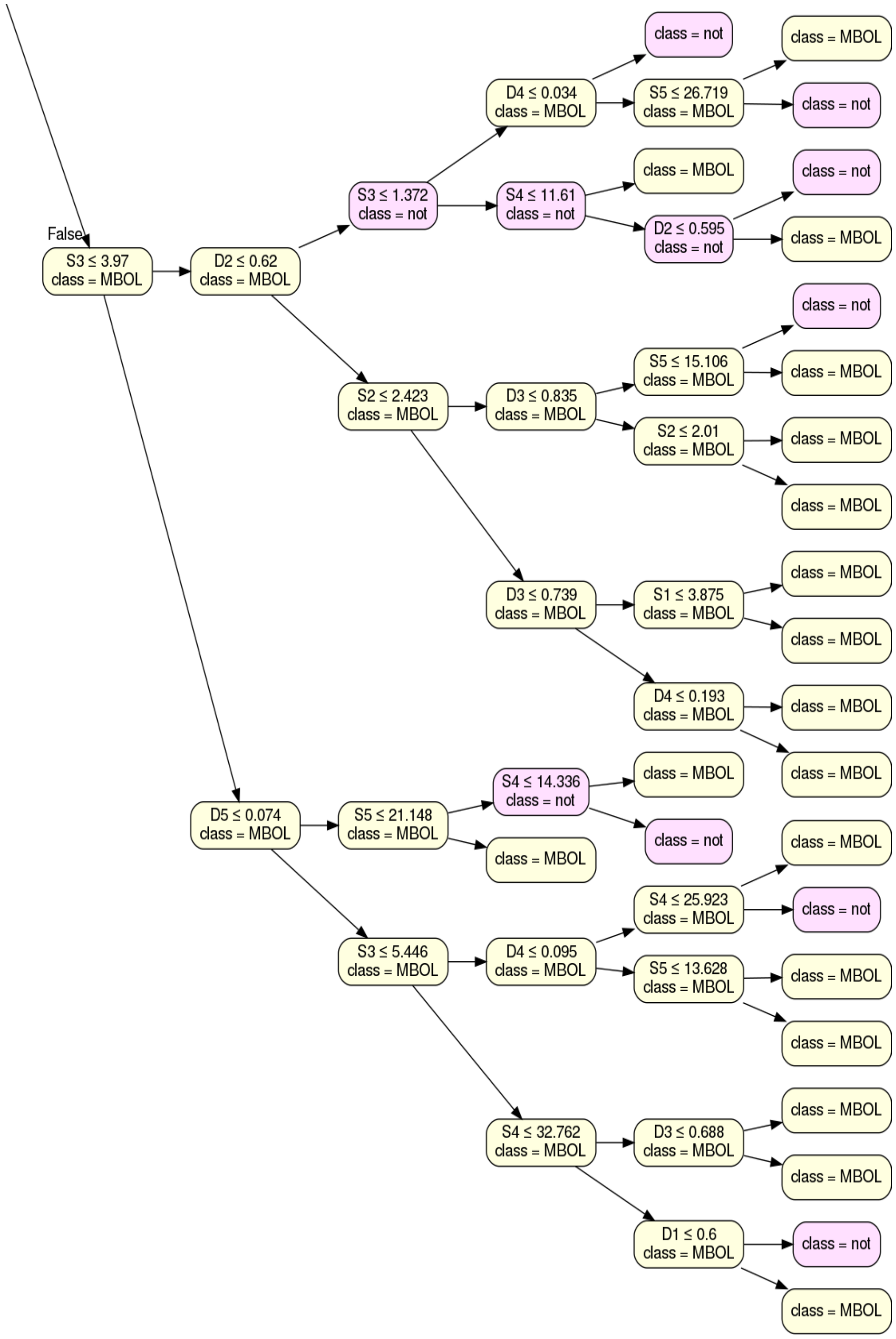

Figure 42 – Decision tree model (part2)

7.2 Simulation modeling of traffic flow

To assess speedup achieved by implementation of special behavior, simulation model of the road has been implemented (figure 43). Model represents agent-based model in continuous space. Model iterates at discrete time steps (30 steps per second).

The agent logic was implemented as follows:

1. The mode of operation of the traffic light is modeled by static agents (with speed equal to 0), which have a switchable presence parameter (enabled/disabled).
2. The controlled parameter of the model is the traffic density. It is supported from the outside: at a density level below the specified one for the lane - a new car appears on the lane; if it is higher than the specified one – new car does not appear. For example, the mean density values for the MBOL-related time blocks is (0.7, 0.8, 0.7, 0.15, 0.2).

Special transport implements the possibility of changing lanes when driving in the oncoming lane. Switching logic: calculate the distance to the cars ahead in two lanes. Once every 3 seconds, special vehicle can change lanes to where the oncoming car is located farther from the ambulance. Check for occupancy on neighbor lane also performed using position and length of vehicles on the neighbor lane. Speed reduced twice when performing lane switch. Timeout for switches also implemented – it can be performed not more than once in 3 seconds.

The general logic of the behavior of agents: when the distance to the next machine is less than twice the bumper distance (3 meters), reduce the speed by the amount of deceleration multiplied by the fraction of the occupied bumper distance; when the distance to the next car is less than the bumper distance - braking to 0; when the distance is more than twice the bumper distance - acceleration. Acceleration-deceleration balance controlled depending on occupancy of multiplied bumper distance. Upon reaching the exit zone of 105 meters, the car is removed from the road.

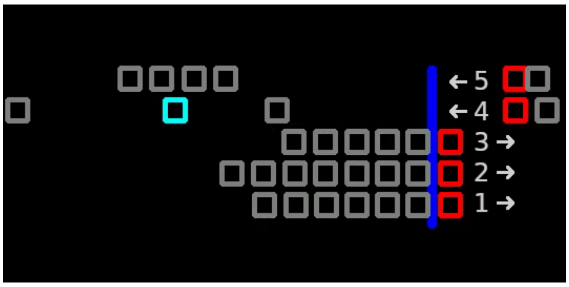

Figure 43 – Visualization of model (grey squares are generic vehicles, blue square – special vehicle, red – traffic light)

Density calculated as ratio of the sum of the lengths of the cars (taken 4.5 meters long) that are present on the road to the length of the road.

As can be seen in figure 44, the model is able to maintain a state of congestion close to the specified.

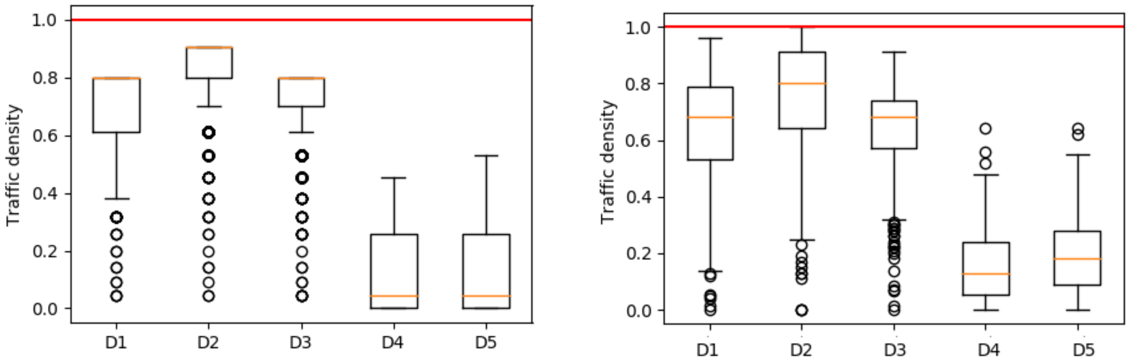

Figure 44 - Measured density in lanes for simulation for an hour (left) and real for MBOL cases (right)

As a method of verification of the system, one can use a comparison of statistical values of an uncontrolled parameter — speed (the system only maintains traffic density). Due to the artificial nature of the system, the velocity variation turned out to be greater than for the actual situation (figure 45).

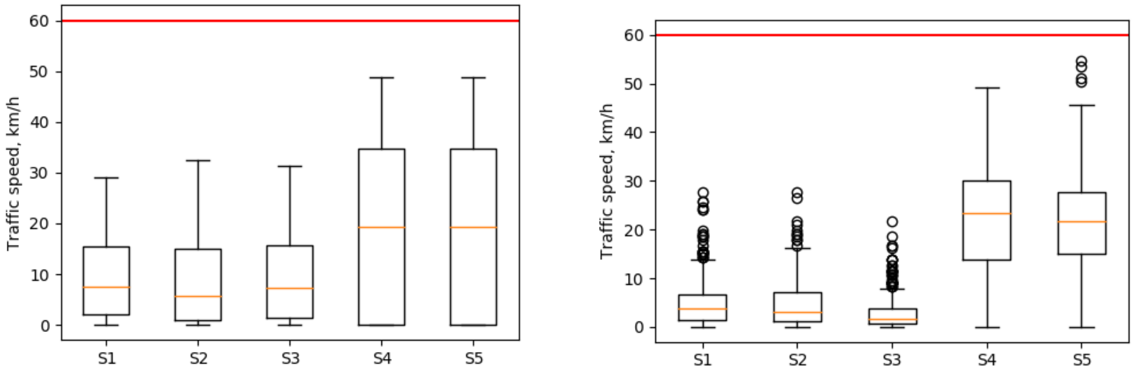

Figure 45 - Measured density in lanes for simulation for an hour (left) and real for MBOL cases (right)

In the conditions of the simulated environment, a comparison was made for 100 exits in lanes 1-3 and 4-5 for cases of normal and MBOL-related condition of the road (figure 46). Two situations are analyzed (congested (related to MBOL) and non-congested), the speedup from the implementation of special behavior in lanes 1-3 (main lanes) and 4-5 (opposite lanes) is analyzed. The average travel time for a MBOL-related road condition for lanes 1-3 and 4-5, is 49 and 35 seconds, respectively (speedup is 1.4).

The travel time on lanes 4–5 does not practically differ due to the small difference in density parameters, however, it can be longer than when driving in a normal lane, due to the need for braking and changing lanes in the presence of oncoming traffic. Lanes 1-3 is characterized by a traffic light mode, so the travel time of a special vehicle in a traffic flow is in a wide range from 15 to 60 seconds, depending on whether the car has time to pass without encountering a red traffic light.

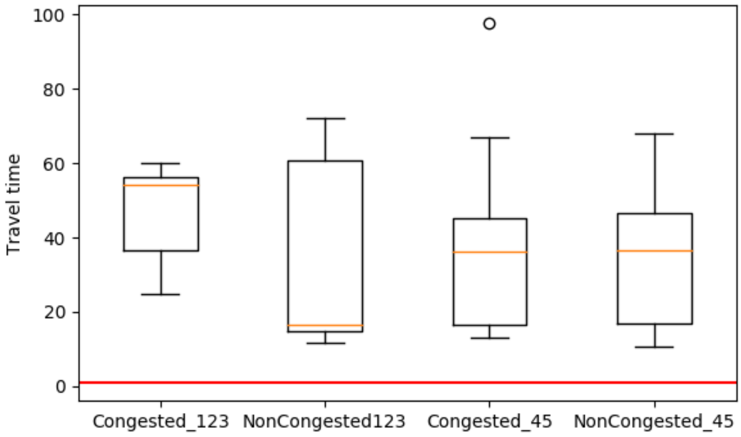

Figure 46 - Travel time of a special transport in a simulated model when moving along lanes 1-3 and 4-5 (oncoming) with traffic density parameters typical for normal (non-congested) and MBOL-related traffic states

Traffic density data can be received from the traffic camera with 20 seconds interval. Classification of a road state can be performed with high accuracy producing recommendation of the fact that MBOL can be performed. Simulation model then can be used to estimate speedup which can be obtained by performing MBOL by the special vehicle driver.

CONCLUSION

A system for collecting and analyzing video data was implemented, which allows the detection of cases of special behavior (MBOL-cases) and the analysis of the traffic situation regarding traffic speed and traffic density using computer vision methods.

25 cases of special behavior were detected, and an analysis of road traffic parameters related to these cases and normal cases (with week duration of observations) was carried out. For hours related to the MBOL-cases, traffic density and speed were estimated.

Neural networks architectures for real-time segmentation were compared, the applicability of these architectures to density estimation and traffic counting was evaluated. Asymmetric real-time segmentation neural architectures were tested. According to the results, the EDANet architecture demonstrates acceptable accuracy and segmentation speed, as well as high learning speed and a small number of parameters (less than a million). Viola-Jones method performed best in terms of quality and speed of recognition in comparison with other object detection methods. Method was used in detect-and-track approach in estimation of traffic flow for MBOL-related hours.

Density and speed estimation if traffic flow allowed to construct a classifier for the MBOL-cases with lane and 20-second blocks precision for the 7 days of observations with mean F1-score equal 0.88. Decision Tree model of depth 6 has also been constructed (with lower mean F1-score 0.39). Simulation model has been implemented to assess possible speedup based on traffic conditions for MBOL-cases of special behavior. Special behavior modeling can be used in construction of decision-support system, which uses acquisition of traffic state from traffic cameras and performs route planning with possible speedup calculation and assessment of possibility of special behavior.

Adaptive suppression for anomaly detection and TrackGrid methods were implemented. These methods were applied to the QMUL Junction dataset, which contains anomalies in traffic flow. The AUC result (0.746) was comparable to the results of other studies using anomaly detection methods.

LIST OF ABBREVIATIONS AND CONVENTIONS

MBOL – movement by the opposite lane.

EDANet - Efficient Dense modules with Asymmetric convolution neural network architecture for image segmentation.

ROC - receiver operating characteristic.

AUC – area under ROC curve.

S1-S5 – estimated traffic speed for lanes 1 to 5.

D1-D5 – estimated traffic density for lanes 1 to 5.

KNN - k-nearest neighbors algorithm, non-parametric method used for classification and regression.

UMAP - Uniform Manifold Approximation and Projection , is a new dimension reduction method, based on the use of Riemann geometry and algebraic topology.

GLOSSARY

TrackLoop – novel method (and graphical feature) for scene dynamics estimation.

TrackGrid – uniformly distributed TrackLoops over image.

Adaptive Suppression – method for anomaly detection, which receives hypothesis about percent of points considered anomalous and relies on iterations of cross-suppression operations and distance metric between data points.

Traffic flow – number of vehicles, which pass over segment of the road in a period of time.

Traffic density – density of vehicles on the road. Can be defined through road occupancy (percent of road area used by vehicles).

Traffic speed – average speed of vehicles in a specified area of the road during specified time interval.

Traffic count – number of vehicles present on the road at the moment in time.

REFERENCES

1. Centers for disease control and prevention [Electronic resource] // Stroke Facts. URL: https://www.cdc.gov/stroke/facts.htm.

2. Saver J.L. Time Is Brain--Quantified // Stroke. 2006. Vol. 37, № 1. P. 263–266.

3. Assis Z.A., Menon B.K., Goyal M. Imaging department organization in a stroke center and workflow processes in acute stroke // European Journal of Radiology. 2017. Vol. 96. P. 120–124.

4. Menon B.K. et al. Optimal workflow and process-based performance measures for endovascular therapy in acute ischemic stroke: Analysis of the solitaire FR thrombectomy for acute revascularization study // Stroke. 2014. Vol. 45, № 7. P. 2024–2029.

5. Spence R.T. et al. Comparative assessment of in-hospital trauma mortality at a South African trauma center and matched patients treated in the United States // Surg. (United States). 2017. Vol. 162, № 3. P. 620–627.

6. Lam S.S.W. et al. Factors affecting the ambulance response times of trauma incidents in Singapore // Accid. Anal. Prev. 2015. Vol. 82. P. 27–35.

7. Gayathri N., Chandrakala K.R.M.V. A novel technique for optimal vehicle routing // Electronics and Communication Systems (ICECS), 2014 International Conference on. 2014. P. 1–5.

8. Créput J.-C. et al. Dynamic vehicle routing problem for medical emergency management // Self organizing maps-applications and novel algorithm design. InTech, 2011.

9. Ibri S., Nourelfath M., Drias H. A multi-agent approach for integrated emergency vehicle dispatching and covering problem // Eng. Appl. Artif. Intell. 2012. Vol. 25, № 3. P. 554–565.

10. Maxwell M.S. et al. Approximate dynamic programming for ambulance

redeployment // INFORMS J. Comput. 2010. Vol. 22, № 2. P. 266–281.

11. Haas O.C. Ambulance response modeling using a modified speed-density exponential model // IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC. 2011. P. 943–948.

12. Tereshchenko S.N., Zhirov I. V. Treatment of the acute coronary syndrome with ST segment elevation at the pre-hospital care // Ration. Pharmacother. Cardiol. 2016. Vol. 6, № 3. P. 363–369.

13. Gendreau M., Laporte G., Semet F. A dynamic model and parallel tabu search heuristic for real-time ambulance relocation // Parallel Comput. 2001. Vol. 27, № 12. P. 1641–1653.

14. Ahmadi Javid A., Azad N. Incorporating location, routing and inventory decisions in supply chain network design // Transp. Res. Part E Logist. Transp. Rev. 2010. Vol. 46, № 5. P. 582–597.

15. Wing M.G., Eklund A., Kellogg L.D. Consumer-grade global positioning system (GPS) accuracy and reliability // J. For. Society of American Foresters, 2005. Vol. 103, № 4. P. 169–173.

16. GPS Accuracy [Electronic resource] // GPS.gov. URL: https://www.gps.gov/systems/gps/performance/accuracy/.

17. Gao Z., Yu Y. Interacting multiple model for improving the precision of vehicle-mounted global position system // Comput. Electr. Eng. 2016. Vol. 51. P. 370–375.

18. Derevitskii I., Kurilkin A., Bochenina K. Use of video data for analysis of special transport movement // Procedia Comput. Sci. Elsevier, 2017. Vol. 119. P. 262–268.

19. Guido G. et al. Evaluating the accuracy of vehicle tracking data obtained from Unmanned Aerial Vehicles // Int. J. Transp. Sci. Technol. Elsevier, 2016. Vol. 5, № 3. P. 136–151.

20. Sundström A., Albertsson P. Self- and peer-assessments of ambulance drivers’

driving performance // IATSS Res. 2012. Vol. 36, № 1. P. 40–47.

21. Sochor J., Juránek R., Herout A. Traffic surveillance camera calibration by 3D model bounding box alignment for accurate vehicle speed measurement // Comput. Vis. Image Underst. 2017. Vol. 161. P. 87–98.

22. Nazare Jr. A.C., Schwartz W.R. A scalable and flexible framework for smart video surveillance // Comput. Vis. Image Underst. 2016. Vol. 144. P. 258–275.

23. Chen Z., Ellis T., Velastin S.A. Vehicle detection, tracking and classification in urban traffic // Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on. 2012. P. 951–956.

24. Yin Z., Cao L. Traffic parameters detection based on video virtual-loop technologies // Applications of Advanced Technologies in Transportation (2002). 2002. P. 885–893.

25. Liu F., Zeng Z., Jiang R. A video-based real-time adaptive vehicle-counting system for urban roads // PLoS One. Public Library of Science, 2017. Vol. 12, № 11. P. e0186098.

26. Temiz M.S., Kulur S., Dogan S. Real time speed estimation from monocular video // Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. Citeseer, 2012. Vol. 39. P. B3.

27. Farnebäck G. Two-frame motion estimation based on polynomial expansion // Scandinavian conference on Image analysis. 2003. P. 363–370.

28. Xiying L. et al. A Traffic Congestion Detection Method for Surveillance Videos Based on Macro Optical Flow Velocity // ICCTP 2011. 2019. P. 1569–1578.

29. Li X. et al. A Traffic State Detection Tool for Freeway Video Surveillance System // Procedia - Soc. Behav. Sci. 2013. Vol. 96. P. 2453–2461.

30. Toropov E. et al. Traffic flow from a low frame rate city camera // Proceedings - International Conference on Image Processing, ICIP. 2015. Vol. 2015-Decem. P.

3802–3806.

31. Ronneberger O., Fischer P., Brox T. U-net: Convolutional networks for biomedical image segmentation // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2015.

32. Zhang S. et al. Understanding traffic density from Large-ScaleWeb camera data // Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. 2017.

33. Oñoro-Rubio D., López-Sastre R.J. Towards perspective-free object counting with deep learning // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2016.

34. Chattopadhyay P. et al. Counting everyday objects in everyday scenes // Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. 2017.

35. Xie W., Noble J.A., Zisserman A. Microscopy cell counting and detection with fully convolutional regression networks // Comput. Methods Biomech. Biomed. Eng. Imaging Vis. 2018.

36. Badrinarayanan V., Kendall A., Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation // IEEE Trans. Pattern Anal. Mach. Intell. 2017.

37. Lo S.-Y. et al. Efficient Dense Modules of Asymmetric Convolution for Real-Time Semantic Segmentation. 2018.

38. Rezatofighi H. et al. Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression. 2019.

39. Romera E. et al. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation // IEEE Trans. Intell. Transp. Syst. 2018.

40. Yu C. et al. BiSeNet: Bilateral segmentation network for real-time semantic

segmentation // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2018.

41. Viola P., Jones M. Rapid object detection using a boosted cascade of simple features // Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. 2001. Vol. 1. P. I--I.

42. Dalal N., Triggs B. Histograms of oriented gradients for human detection // Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005. 2005. Vol. I. P. 886–893.

43. Cheung S.-C.S., Kamath C. Robust techniques for background subtraction in urban traffic video // Proceedings of SPIE. 2004. Vol. 5308, № 1. P. 881–892.

44. Fu H. et al. A vehicle classification system based on hierarchical multi-SVMs in crowded traffic scenes // Neurocomputing. 2016. Vol. 211. P. 182–190.

45. Mandellos N.A., Keramitsoglou I., Kiranoudis C.T. A background subtraction algorithm for detecting and tracking vehicles // Expert Syst. Appl. 2011. Vol. 38, № 3. P. 1619–1631.

46. Guan W., He S. Statistical features of traffic flow on urban freeways // Phys. A Stat. Mech. its Appl. 2008. Vol. 387, № 4. P. 944–954.

47. Lee G., Mallipeddi R., Lee M. Trajectory-based Vehicle Tracking at Low Frame Rates // Expert Syst. Appl. Tarrytown, NY, USA: Pergamon Press, Inc., 2017. Vol. 80, № C. P. 46–57.

48. Samuel O.W. et al. Multi-technique object tracking approach- A reinforcement paradigm // Comput. Electr. Eng. 2018. Vol. 66. P. 557–568.

49. Kalal Z., Mikolajczyk K., Matas J. Forward-backward error: Automatic detection of tracking failures // Proceedings - International Conference on Pattern Recognition. 2010. P. 2756–2759.

50. Mithun N.C., Howlader T., Rahman S.M.M. Video-based tracking of vehicles using

multiple time-spatial images // Expert Syst. Appl. 2016. Vol. 62. P. 17–31.

51. Kaltsa V. et al. Swarm-based motion features for anomaly detection in crowds // 2014 IEEE International Conference on Image Processing, ICIP 2014. 2014.

52. Pathak D., Sharang A., Mukerjee A. Anomaly localization in topic-based analysis of surveillance videos // Proceedings - 2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2015. 2015.

53. Fu Z., Hu W., Tan T. Similarity based vehicle trajectory clustering and anomaly detection // Proceedings - International Conference on Image Processing, ICIP. 2005.

54. Chan A.B., Vasconcelos N. Modeling, clustering, and segmenting video with mixtures of dynamic textures // IEEE Trans. Pattern Anal. Mach. Intell. 2008.

55. Li W., Mahadevan V., Vasconcelos N. Anomaly detection and localization in crowded scenes // IEEE Trans. Pattern Anal. Mach. Intell. 2014.

56. Mahadevan V. et al. Anomaly detection in crowded scenes // Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. 2010. P. 1975–1981.

57. Jiang F., Wu Y., Katsaggelos A.K. Detecting contextual anomalies of crowd motion in surveillance video // Proceedings - International Conference on Image Processing, ICIP. 2009.

58. Kelathodi Kumaran S., Dogra D., Roy P. Anomaly Detection in Road Traffic Using Visual Surveillance: A Survey. 2019.

59. Chaker R., Aghbari Z. Al, Junejo I.N. Social network model for crowd anomaly detection and localization // Pattern Recognit. 2017.

60. Liu F.T., Ting K.M., Zhou Z.H. Isolation forest // Proceedings - IEEE International Conference on Data Mining, ICDM. 2008.

61. Ionescu R.T. et al. Detecting abnormal events in video using Narrowed Motion Clusters // arXiv Prepr. arXiv1801.05030. 2018.

62. Wold S., Esbensen K., Geladi P. Principal component analysis // Chemom. Intell. Lab. Syst. Elsevier, 1987. Vol. 2, № 1–3. P. 37–52.

63. Maaten L. van der, Hinton G. Visualizing Data using t-SNE // J. Mach. Learn. Res. 2008.

64. McInnes L. et al. UMAP: Uniform Manifold Approximation and Projection // J. Open Source Softw. 2018.

65. Hsieh C.-Y., Wang Y.-S. Traffic situation visualization based on video composition // Comput. Graph. 2016. Vol. 54. P. 1–7.

66. Wan Y., Huang Y., Buckles B. Camera calibration and vehicle tracking: Highway traffic video analytics // Transp. Res. Part C Emerg. Technol. 2014. Vol. 44. P. 202–213.

67. Chen H.H. Vision-based tracking with projective mapping for parameter identification of remotely operated vehicles // Ocean Eng. 2008. Vol. 35, № 10. P. 983–994.

68. Henriques J.F. et al. High-speed tracking with kernelized correlation filters // IEEE Trans. Pattern Anal. Mach. Intell. IEEE, 2015. Vol. 37, № 3. P. 583–596.

69. Ali A. et al. Visual object tracking—classical and contemporary approaches // Frontiers of Computer Science. 2015. Vol. 10.

70. Wang L., Chen F., Yin H. Detecting and tracking vehicles in traffic by unmanned aerial vehicles // Autom. Constr. 2016. Vol. 72. P. 294–308.

71. Верещагина Л. Как устроен «Яндекс.Навигатор» [Electronic resource] // The Village. 2017. URL: https://www.the-village.ru/village/business/how/278022-navigator.

72. Stenovec T. Google has gotten incredibly good at predicting traffic — here’s how // Tech Insid. 2015.

73. Zhang M. et al. Learning Universal Sentence Representations with Mean-Max

Attention Autoencoder // arXiv Prepr. arXiv1809.06590. 2018.

74. Loy C.C., Xiang T., Gong S. From local temporal correlation to global anomaly detection // The 1st International Workshop on Machine Learning for Vision-based Motion Analysis-MLVMA’08. 2008.

LIST OF ILLUSTRATIONS