International Journal
Information Technology Applications (ITA)
Volume 9, Number 2, December 2020

ABOUT THE POSSIBILITY OF USING ARTIFICIAL INTELLIGENCE TECHNOLOGIES WHEN RECOGNIZING IMAGES OF OBJECTS FROM DRONES

Andrey Preobrazhenski, Oleg Choporov, Artem Lvovich

Abstract:

The problems associated with image recognition have long been solved within various practical tasks. This paper discusses the possibilities of using artificial intelligence technologies for recognizing images of objects from drones. The problem under consideration in this work is of such a type that it is necessary to recognize images of various objects belonging to the same classification and belonging to the same class; a certain number of objects in the image form a scene. In the developed system, a neural network will be used to identify and classify objects in real time from a video stream. The substantiation of the choice of the Caffe library for the system under consideration is given. Several options for constructing a neural network are considered. Empirically, it was found that for the neural network to work correctly, namely to work with the resulting video, the required height is 28.4 meters. After overcoming this height, the video quality deteriorates sharply, which greatly affects the classification of objects. The work of the classifier with other indicators is shown.

Keywords:

Computer network, traffic loading, protocol.

ACM Computing Classification System:

Network protocols, network algorithms, network types.

Introduction

Image recognition is widely used in various aspects of human life, for example, it can be the definition of objects for augmented reality, quality control of the object, definition of obstacles, search for cars and much more.

This paper proposes to use the technology of convolutional neural networks to recognize images received from a camera located on a drone.

The target audience of the system being developed is private users and individuals. Opportunities when using the developed system:

- the ability to determine the scale and type of objects by the drone, change the trajectory based on the data received;
- generation and viewing of a report of the distance traveled and identified objects;
- the ability to work on mobile devices;
- the ability to work on other devices equipped with a video camera and sufficient computing power.

Thus, a natural requirement for the system arises. This requirement, due to the technological features of mobile devices, imposes a number of restrictions on the technologies used, as well as the system architecture.

1 Features of Solving the Problem of Object Recognition from Drones

Drones, or as they are also called, copters, are actively gaining popularity not only among fans of video filming, but also among business structures. Copters take pictures of the area, spray fertilizer in the fields and even paint, in other words, they do all kinds of work. Most of the actions are not done manually, but from the control panel, or performed using the program.

To save money during production, drone manufacturers use cheap components, which entails weak computing capabilities, critically low battery power, no USB inputs for writing their own algorithm, and much more.

There are different kits of drones on the market that you can choose to suit your needs. It is worth noting that there are specialized drones on the market for working with large amounts of data and for performing calculations, the purchase of which can only be purchased by business structures.

For commercial purposes, drones are equipped with good payload, flying elements, the ability to geolocate, work via wireless channels such as Wi-fi or Bluetooth, and an interface for writing programs.

Mobile devices are an essential part of any person. There are many different types of gadgets in the world. Unlike full-size computers, some mobile devices do not have the ability to use powerful graphics modules that will parallelize computations over floating point numbers. The only available graphics processing capability is the compact and cold graphics coprocessors that have limited computational capabilities [1].

Another limitation of mobile platforms is the limited battery life. In this case, it is important not only the operating time length without connecting to the network, but also the intensity of energy consumption. If the battery is discharged too quickly, there is a risk of degradation of the energy cells due to heating.

The third limitation is networking. In unintended places, there is a risk of signal loss and disconnection from cloud resources, which allow shifting computational operations and processing of output data.

The problem under consideration in this work is of such a type that it is necessary to recognize images of various objects belonging to the same classification and belonging to the same class (obstacle, subject, etc.), a number of objects in the image form a scene. If you look for comparisons, then the task being performed can be compared with locating a location in geodesy. In geodesy, any land operation requires a number of documents and expert opinions. It is required to identify objects on the territory, to outline the boundaries. There are no obstacles for a research drone operating in the air, due to this, objects can be identified as accurately as possible. Based on the footage, specialists can make maps for future projects. One of the approaches is the development of an algorithm based on a neural network that can group and classify objects (trees, stones, old buildings, etc.) based on the received video stream from a flying drone, saving time and effort for humans.

In our case, the task is to recognize objects, any obstacles, barriers, to determine the silhouettes of people, as well as the ability to determine a specific scene and those close to it by the same image parameters.

2 Features of the Characteristics of the Applied Convolutional Neural Network

In this system, a neural network will be used to identify and classify objects in real time from a video stream. Thus, the critical conditions for the operation of the system are the following characteristics of the network: the network has a pronounced activation peak on the video stream with the object for which it was trained; and there are no false activations on adjacent and nearby scenes in the panorama.

It is also desirable that the neural network has a minimum or no activations outside the target range, since this positively characterizes the network, allowing to reduce the load on the service modules of the system that perform classification on a mobile device [2].

Below are some libraries on the basis of which the recognition process can be conducted. Fig. 1 shows the architecture of the formed neural network.

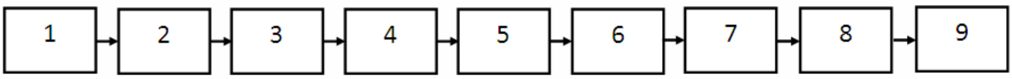

Fig.1. Neural network architecture.
1 - input, 2, 4, 6 - convolutional layers, 3,5 - subsample layers,
7 - a layer of ordinary neurons, 8 - output layer, 9 - image classes

Caffe is a popular library for convolutional neural networks. It was developed by the Berkeley Vision and Learning Center (BVLC). The high speed of work is withered. It is executed on both CPU and GPU. The algorithm is written in C ++, but has Python and Mathlab wrappers.

Deeplearning4j is a Java and Scala library that uses an open source framework to implement Apache Spark's distributed processing of unstructured and semi-structured data. It is a general purpose deep learning library designed to run in a JVM environment. The core of the library is a scientific computing block that is written in C ++. Allows to create layers with specified parameters. Integrated into Hadoop and Kafka packages. Deeplearning-hs is a deep learning library in the Haskell language that supports distributed computing on CUDA technology. MatConvNet is an implementation of convolutional neural networks in MATLAB. Neon - announced by the developers as the fastest framework for convolutional neural networks and deep learning with support for computing on GPU and CPU. The front-end is made in Python, while the algorithms themselves are implemented in a specially developed shader assembler. Developed by Nervana Systems, which was acquired by Intel. TensorFlow is a library from Google that is licensed under the Apache 2.0 license. It supports calculations on CPU, GPU and specially developed by Google TPU (Tensor proceqqing units). Front-end is in Python. It is a popular, well-documented and well-developed library.

Theano is a deep learning library for Python with APIs (for the most part) compatible with the popular NumPy library.

The user can write symbolic formal mathematical expressions from which derived code is automatically generated. Hence, the user does not need to program gradients or back propagation of the error. Such expressions are automatically compiled into shader code for CUDA, to optimize calculations on the GPU.

Torch (torch.ch) is a scientific computing framework with broad machine learning support written in C and Lua. This framework is currently used by the research unit in the field of artificial intelligence of Facebook, as well as by Twitter to implement systems for automatic classification of user-generated content [3].

The main approach is to write a highly efficient library core that executes the algorithm itself in a low-level language (C, C ++, Python). Then, to ensure usability and readability of the code, the user is presented with a high-level frontend from where, using a more convenient syntax, calls to the library core can be made [4]. The following criteria for evaluating the technologies used can be formulated:

- The shell of the library used for image recognition must be written in a programming language, the code of which can be executed on a mobile device and supported by the drone system;
- The core of the library should be easy (without long additional configuration) to compile for processors of the ARM architecture;
- The kernel must effectively use all the capabilities of modern GPUs for parallelizing computations on the training server;
- The technology must be popular, supported, documented and have a developed active developer community;
- Ability to control the aircraft (drone) from the phone.

Table 1. Compliance of popular libraries with evaluation criteria.

Library (framework)	Easy to launch the shell on a mobile device	Easy of building the core for ARM	Optimization for GPU	Popularity
Caffe	-	+	+	+
Deeplearning4j	+	-	+	+
deeplearning-hs	-	-	+	-
MatConvNe	-	-	+	+ -
Neon	-	-	+	+
TensorFlow	+	-	+	+
Theano	-	-	+	+
Torch	-	-	+	+

All this is necessary for ease of obtaining technical advice on the implementation of a demonstration project.

From (Tab.1) you can see that none of the options meets all the criteria. In order for the project to be successfully completed, you will need to independently supplement the technical solution with the necessary tools. The last step is the selection of criteria that can be implemented with less effort.

It is undoubtedly easier to add a wrapper to a kernel that builds well on the target platform than patching a kernel written in a low-level language for another platform.

As a result of the evaluation and analysis, it was decided to use the Caffe library. To complete the project, it was required to execute the library frontend in a high-level language and implement the call mechanism into the library core code.

Neural network training is directly dependent on properly selected training data. Poor data or misrepresentation of data will lead to incorrect results. Submitting one image to the input, for example, an image of a car, does not give complete confidence that now the neural network will easily detect all cars.

In order to identify an object without errors, thousands of test cases are needed to be processed, it is also necessary to select the edges of images (the so-called silhouette of an object, consisting of digital values and stored in a text document). (Fig.2) shows a visual representation of ANN training.

The first stage of training is a convolution layer, another name is feature maps, at this stage the existing features in the image are highlighted (a 5x5 kernel runs over the entire image and there is a matrix from 0 to 255 corresponding to the color of pixels).

The second stage of training (Fig.3). If at the previous stage of convolution detailed features of the image were revealed, then for further processing such a detailed image is no longer required, it is compressed to a less detailed one. In addition, filtering out unnecessary parts helps not to re-train. Then there is one more layer of convolution to highlight new features.

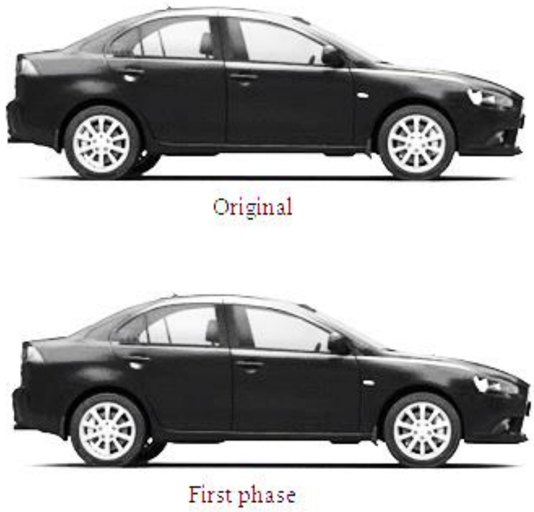

Fig.2. Learning object detection.

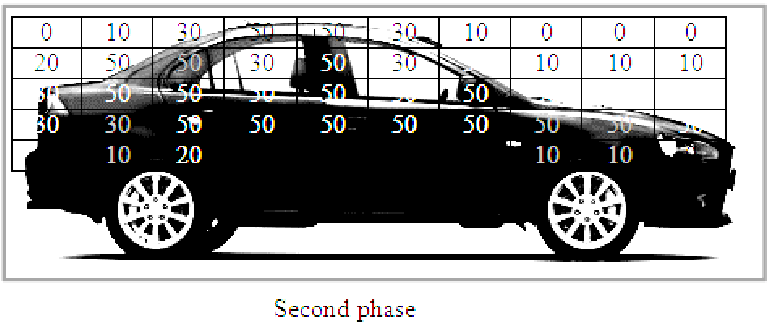

Fig.3. Second stage of training.

Third stage of training. Attaching Fully Connected Layers is all about referring to the output of the previous layer and defining properties that are more related to a specific class. In other words, if the program predicts that there is a car in the image, the property maps that reflect high-level characteristics such as 4 wheels, body, 4 doors, etc. should have high values. The fully connected layer looks at the fact that high-level features are strongly associated with a certain class and have certain weights.

The last stage of training is the output layer, the stage at which the ANN gives an answer to what is located in the image.

The training of the neural network has not finished yet, since only one example has been demonstrated, the network may well define the car object, but only the one in the example, other cars have a different structure and different features. The complete algorithm of the neural network will be demonstrated below.

For the first training of a convolutional artificial neural network, a data set was manually formed, consisting of images obtained on specialized torrent trackers. The advantage of these torrent trackers is that the image is initially divided into categories, which greatly reduces the amount of manual moderation. (Tab.2) contains image samples for ANN training.

Table 2. Sample of images for training ANN.

	Valid images	Not valid images	Valid images with deliberate modification
Trees	1080	1201	448
Stone barriers	1361	1480	396
Cars	1516	1906	243

When forming the training set, 3 categories of images were identified for comparative analysis - valid, invalid and valid images with deliberate change. The category of valid images includes photographs of people, images of cars, trees, large stones. Photos of poor quality (blurring, etc.) or photos that do not contain the desired object are invalid. Valid images with deliberate modification are a category of valid images in which random patterns were applied to the desired object using the Paint application.

Any change in the image, even a minor one, should be perceived by the neural network as a new drawing, so we can check the network's learning ability. The main goal of the study is not to determine such detailed parameters as the make and model of the car, type and type of tree, the main goal is to determine the classification possibilities [5, 6].

Solving a specific problem is an iterative process consisting of the following steps:

1. Designing network architecture;
2. Realization of the network;
3. Network setup;
4. Network training;
5. Testing the network.

If the test results are completely unsatisfactory (the network is retraining, the network cannot find the local maximum, continuing to change the weights in the range of values, etc.), then a new iteration is resumed from step 1.

If the network reacts to the data, but the accuracy and error do not allow achieving the target parameters, we can assume that the architecture is selected and a new iteration is resumed from step 3

To determine the criteria for testing a network, it is required to describe the scenario of its use.

Thus, a sufficient condition for the network to work is the following - the value of the network output on frames with a recognizable object should be the largest among all images presented to it.

Also, the criterion for the quality and applicability of the network will be the following condition: the next largest values of the network output after the values corresponding to the recognized object should be separated from the latter by a statistically significant, well-distinguishable distance, which will reliably distinguish the activation of the network on the object from the rest of the weaker activations.

The behavior of the network on the overview file depends not only on its architecture and settings, but also on the training examples that were presented to it at the training stage. Empirically, it was found that all tested networks learn better in some examples.

To train the neural network on real images, we selected videos filmed with the aircraft used in this work and a mobile phone. It was decided to use the fragments filmed in the central park "Dynamo" in Voronezh in several locations in different weather and time of day.

3 Research Results

To test each of the 4 options for neural network architectures, the following sequence of steps was used:

1. Splitting the corresponding survey video files into frames and defining the frame range in which the object appears, which the tested network was trained to recognize;
2. Testing the network on survey video files checking for the presence of the most pronounced peak in a certain range at the first stage;
3. Checking that there are no higher peaks outside this range;
4. Fixation for subsequent analysis of lower, but pronounced peaks outside the range, although they were not false network activations;
5. Testing the network on its own training set of images to check for failures;
6. Checking the network on files captured in other locations, with a large number of false activations or suspicion of retraining, to clarify whether the activation correlates with location signs (clouds, paving stones, the shape of windows in cars) or the network is, in principle, activated on the widest image classes (Tab.3).

The quadcopter has several important functions:

- Follow me - a function that allows you to follow the operator through the use of navigation systems;
- Fly by waypoint - a function that allows you to follow the points indicated before the flight (smartphone required);
- Follow to point - a function that allows you to direct the quadcopter to a specific point (smartphone required);
- Return home.

The quadcopter requires a smartphone. To use the functions, you need to install a free app from PlayMarket for Android or AppStore for iOS.

After connecting the copter to a smartphone, you need to carry out standard settings: authorization, connection to a smartphone, setting the GPS error (for accurate localization). Then in the settings you can add an additional host (server \ PC) for video transmission, but this will significantly affect the performance and battery life.

The laptop of the Lenovo series model IdeaPad310 was chosen as a server:

1) CPU - Intel core i3;
2) RAM - 8GB;
3) Solid State Drive - SEAGET;
4) Video card - Nvidia 910m.
5) Wi-fi module - BCM943142HM 802.11B / G / N + BT4.0 (1 * 1) [0C011-00041300]

Notebook was not subjected to certain settings, it was decided to use the standard settings of the Windows 10 operating system.

To work with the quadrocopter, the QuadraPlan software was installed, the settings were carried out via a mobile phone.

The drone operation algorithm has a list of standard actions:

1 Checking components - checking the connection to the mobile device, as well as the connection to the server, checking the battery charge, the functioning of the controls (blades);
2 Launching a video stream on the server and mobile phone;
3 Takeoff;
4 Checking the presence of a route (in this version it is better to indicate the route in advance, since when the drone is in the air, the check can cause a fatal error);
5 Flight to the specified point;
6 Checking the route - it was decided to check the current location with the planned route every 30 seconds.
7 Signaling of the successful achievement of the set route;
8 Return to the starting point;
9 Landing.
10 Uploading the received data.

Table 3. Network test results.

Network Option	Peak activations (when shooting)	False activations (when shooting)	Other peaks, peaks / frames (frames)	Activations on the training set are below 50%	Activations on data from other locations
1	23	0	4\85	0	2
1	105	2	2\506	0	
2-1	154	2	2\746	0	2
2-1	96	2	2\663	0	
2-2	31	0	2\50	0	0
2-2	72	0	3\183	0	
3	45	0	0	0	0
3	100	2	1\134	0	

It is worth noting that this model is a budget model, so it is severely limited not only in terms of internal indicators, but the possibilities are also severely curtailed, for example, at the peak of a heavy load, the copter was active for only 10 minutes, the range was no higher than 130 meters, going beyond threatened with artifacts on the video, and control failures were also noticed. Also, when the battery was low, the drone stopped responding to signals from the mobile phone. A full description of the drone algorithm is shown in (Fig.4).

The most difficult thing in setting up the drone and the artificial intelligence (AI) module was to teach it to determine where the obstacle (object) is and to maneuver. The ANN came to the rescue, since by classifying the object (after each convolution layer in the image, unnecessary elements are removed), it determines its localization in the image. The image obtained in this way passed through a function written in Python, which changed the resolution of the resulting image to 1280x720, a visual example is shown in (Fig.5).

If the object occupies the area of one of the indicated letters, the drone will maneuver in the opposite direction. The whole essence of the object analysis algorithm for AI is to check the conditions, whether the object has occupied the A - B field, etc.

Empirically, it was found that for the correct operation of the neural network, namely work with the resulting video, the required height is 28.4 meters. After overcoming this height, the video quality deteriorates sharply, which greatly affects the classification of NS objects.

The maximum speed of the drone used is 50 km/h, when this speed is reached and ascending to a height of 28 m, large objects remain distinguishable, such as cars, food stalls, large stones. Objects (People), captured in this mode, acquire the effect of "spots", it is difficult to answer the question of whether they can be classified, since they were not identified during the test.

The viewing angle of the drone's webcam is 180 degrees, so it was decided to build a route, following which the same number of objects would be constantly encountered. The flight was built through a mobile app to reduce the risk of altitude and speed changes. The tests were carried out at the car park of the Dynamo park in Voronezh; cars (20 units), bystanders, a brick fence (2 units), which was identified as an obstacle, were chosen as objects for testing. During the tests, the cars and the fence did not move, the number of people in the frame during some tests increased, or they were absent.

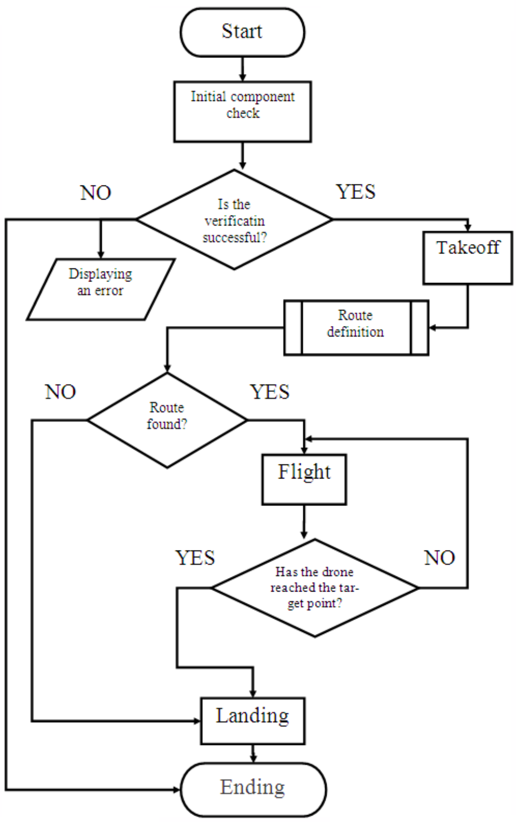

Fig.4. Algorithm of drone actions.

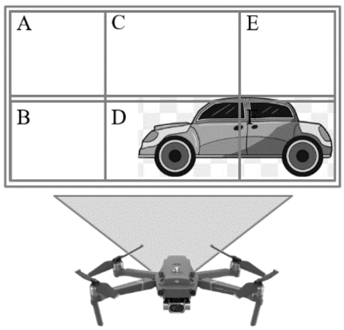

Fig.5. Object definition area.

(Fig.6) shows the route, crosses show the location of the objects (the number of crosses in the figure may not exactly coincide with the objects during testing).

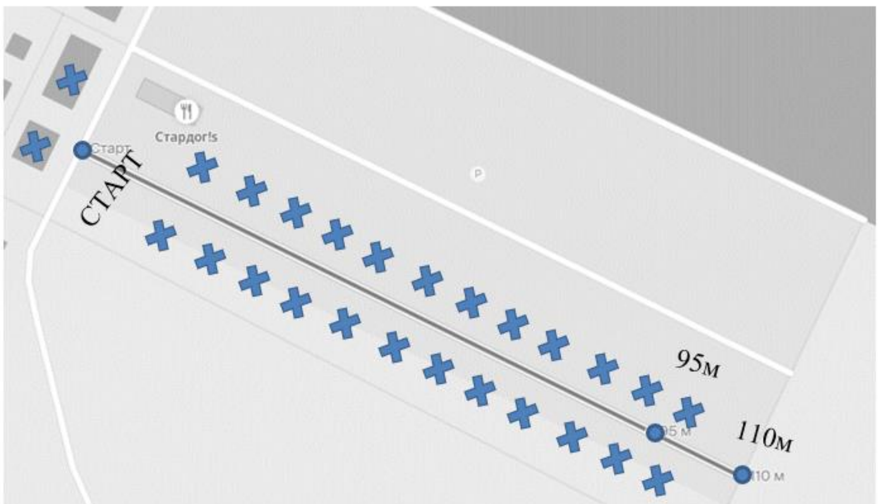

Fig.6. Test area (Park "Dynamo", car parking), starting point on the left side.

The webcam built into the drone records video in 720p (1280x720) quality. The range of the signal from the drone is 130m (in good weather conditions), after the drone crosses the 150m mark, malfunctions are possible. (Tab.4) shows at what indicators the operation of the ANN classifier becomes impossible, where 100% is a complete classification of objects, 0% is the definition of objects is impossible, the drone's performance is impaired [7].

Based on the presented table, we can conclude that the maximum permissible height and speed for the successful operation of the classifier is an altitude of 60-80 m and a speed of 30-40 km/h, at which most of the objects were successfully classified (from 20 to 27 objects out of 30).

Table 4. Work of the classifier with other indicators.

	Speed 0-10 km/h	Speed 10-20 km/h	Speed 20-30 km/h	Speed 30-40 km/h	Speed 40-50 km/h	Speed 50-55 km/h
Height 0-20 m	97.3%	93.7%	87.3%	83.1%	81.1%	73.3%
Height 20-40 m	87.1%	83.9%	80.6%	80.6%	77.4%	77.4%
Height 40-60 m	87.1%	83.9%	80.6%	77.4%	64.5%	61.3%
Height 60-80 m	87.1%	83.9%	77.4%	64.5%	61.3%	58.0%
Height 80-100 m	64.5%	64.5%	58.0%	58.0%	0%	0%
Height 100-280 m	0%	0%	0%	0%	0%	0%

Conclusion

As a result of writing a scientific work, the set goal was achieved - to study the possibilities of using neural networks for recognizing images received from drones.

We analyzed the available libraries and frameworks for designing ultra-precise neural networks that implement deep learning algorithms, searched for a neural network architecture that solves the problem, tested the neural network and conducted tests using a quadcopter.

References

[1] Panevin R.,Yu., Preobrazhensky Yu., P. (2008) Structural and functional requirements for a software package for presenting knowledge. *Bulletin of the Voronezh Institute of High Technologies.* No. 3. S. 061-064.

[2] Preobrazhensky Yu., P., Marenkov N., M. (2020) Analysis of algorithms useful in the formation of augmented reality systems. Bulletin of the Voronezh Institute of High Technologies. *Bulletin of the Voronezh Institute of High Technologies.* No. 1. S. 40-42.

[3] Preobrazhensky Yu., P., Myasnikov N., M. (2020) Analysis of the prospects of information technology in the field of Internet of Things. *Bulletin of the Voronezh Institute of High Technologies.* No. 1. S. 43-45.

[4] Pustylnik I.,E., Preobrazhensky Yu., P (2019) Securing messages between the server and IoT devices. Bulletin of the Voronezh Institute of High Technologies. *Bulletin of the Voronezh Institute of High Technologies.* No. 4. S. 40-45.

[5] Lvovich I.,Ya., Preobrazhenskiy A.,P., Preobrazhensky Yu.,P., Choporov O.,N. (2019) About tracking processes on various devices.
Bulletin of the Voronezh Institute of High Technologies. No. 4. S. 49-51.

[6] Lvovich I.,Ya., Preobrazhenskiy A.,P., Preobrazhensky Yu.,P., Choporov O.,N. (2019) On the quality of the Internet of Things system. *Bulletin of the Voronezh Institute of High Technologies.* No. 9. S. 52-55.

[7] Preobrazhensky Yu., P., Miroshnik D.,N. (2018) Analysis of fuzzy search methods. *Bulletin of the Voronezh Institute of High Technologies.* No. 4. S. 82-84.

Authors

Prof. Andrey Preobrazhenski
Doctor of Sciences (Engineering), Professor,
Voronezh Institute of High Technologies, Voronezh, Russia
E-mail: app@vivtl.ru

Prof. Oleg Choporov
Doctor of Sciences (Engineering), Professor
Voronezh Institute of High Technologies, Voronezh, Russia
E-mail: choporov_oleg@mail.ru

Artem Lvovich
Student
Comenius University in Bratislava, Slovakia
E-mail: artemlv2000@mail.ru