Neural network methods in real-time recognition of road infrastructure objects

M Gorodnichev 1, M Moseva 1, M Yashina 2, Kh Dzhabrailov 1, R Gematudinov 2

1 Moscow Technical University of Communications and Informatics, Moscow, Russia
2 Moscow State Automobile and Road Technical University, Moscow, Russia

E-mail: m.g.gorodnichev@mtuci.ru

Abstract. The purpose of the work is to study the typification methods of moving objects based on a neural network. It turned out that the use of image classifications is so widespread today that it is difficult to imagine a sphere in which any algorithms are not implemented. The task of the work is developing system for detecting road signs on the video.

1. Introduction
Replacing many functions performed by humans with automated systems requires the development of new image recognition technologies. The human brain is a very complex system and the solution at first glance of very simple tasks that a person learns in the process of growth and development is a serious problem.

The standard task posed to a child of five years, consisting in determining the animal: a dog or cat, after a short training is easily solved. But to solve this problem, a person uses thousands of neural connections, and the signs by which the desired animal is detected are not fully formulated, and, therefore, it is not so simple to designate the separation of such objects for the robot [1].

They are trying to exclude an expensive human resource from many routine operations: determining defective products, triggering transport responses associated with the appearance of a certain object on the road or roadside, identifying thefts, tracking the implementation of a certain set of actions, and many others.

Computer vision, which replaces the biological technology of image analysis with the human brain, involves not just fixing the situation in the form of an image, but also performing the analysis seen to obtain data in order to select the optimal action in certain conditions.

The breadth of application of image analysis is also associated with the expansion of the scope of use of controlled robots and systems that will be able to replace a person in special conditions or adjust his activity through tips. Such a system used to recognize a problem on the road, indicate which road sign the driver has passed and will help with recognizing its value. And, therefore, more adequately and most importantly with less time, they will appreciate the possibility of applying the necessary maneuver.

One of the sides of this recognition is the classification of images, which contributes to the simplification of the analysis and the development of a management decision based on it. In this case, images can be included in some video sequence.

The aim of this work is to study the typification methods of moving objects based on a neural network.
To achieve this goal, the work involves solving a whole range of tasks:

- study of the problem of image typing in the aspect of the spheres using implementation technologies;
- analysis of existing algorithms focused on image typing;
- development of a design of a typing system for moving objects;
- conducting an experiment and evaluating the effectiveness of the classification on the example of real data.

The problem of the typification of objects is quite broadly described in scientific sources. Among the main works, it is possible to single out the research by Bui V.Sh., Litvinov Yu.V. [2,3], as well as Isaev A. L., Gazarov D. A., Evseeva S. D. [4].

The work with the video image was carried out by: Shashkov B.D., Shepelev K.V. [5]. The solution to this problem using neural networks involved Buiko A. Yu., Vinogradov A.N. [6], Sikorsky O.S. [7].

An acceptable, ready-made universal solution to such problems is still missing, although many large corporations in the IT sector, in particular Facebook, Google, NVidia, are actively conducting fundamental and applied research in the field of machine learning and computer vision.

2. Image classification task

Category Recognition is still considered quite problematic, despite obvious successes in solving a number of recognition tasks related to the field of computer vision (for example, the task of finding instances of an object (Instance Recognition) or the task of localizing a specific object.

The vast majority of the classification mechanisms currently presented are based on Visual Features. The techniques based on them gave satisfactory results in the 1990s when solving the problems of finding instances of objects and localizing an object. At the beginning of this century, attempts have been made to adapt these mechanisms to solve the problem of image classification. In different classification methods, there is a significant difference in the principles of applying data on visual features, however, there is the possibility of highlighting some general algorithm.

1. Creating a training set of images labeled in accordance with the classes.
2. Extracting visual features from the training set.
3. Modification of the information received for subsequent research.
4. The use of processed data to study the next input image, followed by a decision on which class it can be assigned to.

One of the most commonly used attributes for classification is color. However, despite the demand, most of the color ranges are sensitive to changes in the light source. In this regard, in cases of the inevitability of this effect, it is advisable to use other features that are included in the model of representation of the object.

The task of classifying (categorizing) an object in an image involves the division into two independent subtasks:

- finding on the image of the object and the subsequent selection of areas of interest;
- recognition and categorization of the received object, or area of interest.

The focus of the first subtask is related to the search for objects for categorization (classification). Often there is no information regarding the location, orientation, size, presence and number of targets. In this case, it is necessary to determine previously unknown parameters, without which it is impossible to further distinguish an object, or a region of interest of a local nature [13,14].

The second subtask is focused on working with the entire image and as a result of its decision, it becomes possible to decide which of the number of available classes the image can be assigned. The main (target) task of classification is to form a decisive function. The decisive function for each of the vectors is the correlation with the corresponding class.

Also, when solving the problem of categorization, well-proven ones are used in solving other problems of this type, namely:

- comparison with the existing template;
• finding invariant features for selected segments;
• use of local features.

3. System design and development
The objective of the developed system is the detection of traffic lights and certain traffic signs on the video image of the road, such as “Pedestrian crossing”, “Direction of movement”. To implement this system, Python was chosen and the open library from Google - Tensorflow object detection API. Tensorflow allows you to train your own designed models or choose a ready-made model for training by configuring it for your own needs. It was decided to use Deep Learning neural networks, because at the moment they are the most advanced. [fifteen].

To achieve this goal, it was decided to pay more attention to preparing data before training. In addition, the preparation of a training sample using the affine transformation will significantly reduce the likelihood of retraining the classifier. To prepare the images before training the classifier, an additional module (STN) will be used.

3.1. Markup and initial data analysis
Before starting training, it is necessary to collect and prepare data on which the model will be trained. The main task is to train the classifier of road signs, which means that you need to collect images of road signs and traffic lights. The photographs were collected by dividing video footage of city trips with a car recorder. After collecting a pool of photographs, it is necessary to mark them in a certain way (mark on each frame objects belonging to the classes that our model will learn to recognize). For labeling the data, the Labelling program was used. As a result of the markup, an xml file of a certain structure is created for each image, which contains the coordinates of the marked objects and the objects belong to classes.

In total, 3 classes of objects were allocated - “Pedestrian crossing” (zebra), “Direction of movement” (direction), “Traffic light” (trafficlights).

After marking up the images, they must be divided into 2 parts - data for training and data for testing. The standard ratio is 70/30, where 70% is a training sample and 30% is a test sample. The training sample includes 700 samples for training, a test sample - 300, image parameters - 32x32x3, the number of image classes - 3.

The distribution of the parts of the dataset into classes is approximately equal, this was considered even at the stages of assembly and layout of the dataset, because the presence of a dominant class in the set can negatively affect the final result of training.

To increase the convergence of images, it is advisable to bring each of the images to uniform illumination. To do this, it is necessary to modify their color scheme in grayscale. For this, both the OpenCV library and the special Python scikit - image library is suitable, the installation of which is easy using pip (in the case of using OpenCV, you will have to independently compile it taking into account a large number of dependencies). To normalize the contrast of images, adaptive histogram normalization (CLAHE) can be used. To execute parallelized CLAHE, writing just a few lines of code is necessary.

The parallelization technique is the following algorithm: the sampling is divided into packets with the subsequent processing of each packet separately from the others. After processing all the packages, they are combined into a single set of data back.

Other important problems include the decrease in the likelihood of retraining the neural network caused by the addition of new different samples. In this case, to add new artificial images, it is enough to modify existing images. This is achieved by rotation, affine transformations and mirror reflection. This action can be performed for the entire data set as a whole.

After pre-processing the data and developing all the generators, the data set is ready for analysis. Now you need to complete neural network training.
3.2. Network training

In this case, two sequential convolutional neural networks STN (spatial transformer network) were used. The input data for it are image packets from the generator (subjected to preliminary processing). Further, it focuses on road signs, while the neural network IDSA performs recognition of a road sign on images, the source of which is [17].

The Spatial Transformer Network is one example of the differentiated LEGO modules that underpin the development and improvement of its own neural network. STN, using a trained affine transformation with further interpolation, removes spatial invariance from images. In other words, the main function of STN is to perform such a rotation or resizing of the original image, so that in the end the determination of the required object for the main classifier network is greatly simplified. The STN subsystem can be located in the CNN convolutional neural network, while the subsystem operates mainly offline with training on gradients that come from the main network.

STN is able to function even in special non-trivial cases (for example, an image contains more than one traffic sign). However, its main advantage lies in improving the quality of the classifier (in this case IDSIA).

One of the most significant difficulties that inevitably arise when interacting with convolutional neural networks is the fact that there is too little invariance to the initial data (different scale, background noise, shooting point).

Unfortunately, the insufficient size of the receptive field in the “pooling” of the standard $2 \times 2$ type leads to the fact that the achievement of spatial invariance seems possible only in deep layers that are close to the output layer. In addition, “pulling” cannot ensure the invariance of scale and rotation.

The most acceptable technique to achieve stability of the model to variations of this kind is to pre-configure the data set, which was described above.

This approach is absolutely satisfactory; however, it is desirable to develop a method of preliminary image processing, at a higher level, providing an increase in the accuracy of the classifier of road signs. And here STN is preferred.

Next, the model is built in TensorFlow. As mentioned above, the ultimate design goal is the recognition of moving objects on the road. To accomplish this task, it is necessary to create and train some classifier. There are several solutions - from LeNet to any other type of network such as SOTA. Most preferable is the use of the IDSIA neural network architecture.

3.3. Model for detection objects using CNN

After selecting the network, the STN module is determined and trained, which takes the initial image as input parameters, modifies it with the help of a sampler, and receives a new image (or, if it works in batch mode, a packet of images). The resulting new image is necessary for the classifier.

It should be noted that the STN module can be easily removed from the computation graph. Instead, a conventional packet generator can be used. In this case, a regular network classifier is formed.

Thus, it is necessary to implement the transformation of source images using STN and submit them to the IDSIA classifier, which performs logit calculations.

After performing the calculation of logits, it is necessary to carry out the optimization of the loss function (the function of losses is the cross-entropy or log loss - this choice is standard in solving classification problems).

After that operations (ops) of optimization and training are set. The purpose of these operations is to propagate errors back to the initial input layers of the network.

It is more expedient to initialize the network with a sufficiently large value of the learning rate (0.2). With this value, the dissemination of information by gradients to the LocNet STN located in the outer layers of the entire neural network is faster. If this is not done, the training of the neural network will take place more slowly (in this case, the so-called “vanishing gradient” problem has a great influence on the speed). If the learning rate is small, the neural network is not able to efficiently implement small road objects in the image.
The applied structure of the classifier is presented in table 1.

| Layer | Name | Parameters | Input | Output |
|-------|------|------------|-------|--------|
| 1     | Convolutional (batch normalization, relu, dropout) | Kernel, 7 x 7, 100 filters | 32 x 32 x 1 (In a set of 256) | 32 x 32 x 100 |
| 2     | Max Pooling | - | 32 x 32 x 100 | 16 x 16 x 100 |
| 3     | Convolutional (batch normalization, relu, dropout) | Kernel, 5 x 5, 150 filters | 16 x 16 x 100 (In a set of 256) | 16 x 16 x 150 |
| 4     | Max Pooling | - | 16 x 16 x 150 | 8 x 8 x 150 |
| 5     | Convolutional (batch normalization, relu, dropout) | Kernel, 5 x 5, 250 filters | 16 x 16 x 100 (In a set of 256) | 16 x 16 x 150 |
| 6     | Max Pooling | - | 8 x 8 x 250 | 4 x 4 x 250 |
| 7     | Optional pooling for multiscale features | Kernels: 8, 4, 2 for layers 1, 2, 3 | - | Features vector 400 + 600 + 1000 |
| 8     | Getting and concatenating features into a multiscale feature vector | - | 2 x 2 x 100 | 2 x 2 x 150 = 2000 |
| 9     | Fully – connected (batch normalization, relu, dropout) | - | 200 attributes (In the set of 256), 300 neurons | - |
| 10    | Logits (batch normalization) | - | 300 attributes | Logits (3 classes) |

As mentioned earlier, in accordance with the operation algorithm, the results of the activation functions of each of the convolutional layers are combined into one vector. It is this vector that arrives at the input of layers that are fully connected. This scheme can be considered as an example of multiscale - features, the purpose of which is an additional improvement in the quality of functioning of the classifier.

It should also be noted that the input parameters for conv1 is an already modified STN image.

Among all the developed TensorFlow models, it is advisable to choose the implementation of STN, which will be used in the developed network.

The main task is to identify and train LocNet, provide the transformer with relevant \( \theta \) values, and insert the STN module into the TensorFlow structure. The transformer is entrusted with the task of generating a mesh and providing modification and interpolation.

LocNet configuration parameters are presented in table 2.

| Layer | Name | Parameters | Input | Output |
|-------|------|------------|-------|--------|
| 1     | Max Pooling | - | 32 x 32 x 1 | 16 x 16 x 1 |
| 2     | Convolutional (batch normalization, relu dropout) | Kernel, 5 x 5, 100 filters | 16 x 16 x 1 (In set of 256) | 16 x 16 x 100 |
| Layer | Name | Parameters | Input | Output |
|-------|------|------------|-------|--------|
| 3     | Max Pooling | - | 16 x 16 x 100 | 8 x 8 x 100 |
| 4     | Convolutional (batch normalization, relu dropout) | Kernel, 5 x 5, 200 filters | 8 x 8 x 100 (In set of 256) | 8 x 8 x 200 |
| 5     | Max Pooling | - | 8 x 8 x 200 | 4 x 4 x 200 |
| 6     | Additional pooling for multiscale | Kernels: 4,2 for convolutional layers 1,2 | - | - |
| 7     | Getting and concatenating into a vector multiscale | - | 2 x 2 x 100; 2 x 2 x 200 | Features vector 400 + 800 = 1200 For fully connected layers |

Fully - connected layers LocNet

| Layer | Name | Parameters | Input | Output |
|-------|------|------------|-------|--------|
| 8     | Fully - connected (batch normalization, relu dropout) | - | 1200 attributes (In set of 256), 100 neurons | - |
| 9     | 2 x 3 matrix θ, defining the affine transformation. Weights are determined by zero values, a matrix similar to the unit one, flowing on the main diagonal of the unit, acts as a free term: \([1.0,0,0], [0,1,0,0]\). | - | - |
| 10    | Transformer. It is a grid generator and a sampler implemented in spatial_transformer.py. The purpose of this layer is to reproduce images that have the same dimensions as the original images (32 x 32 x 10) using the affine transformation (therefore, the result is a rotated or close-up image). | - | - |

By structure, LocNet convolutional layers are similar to IDSIA with the difference that LocNet includes two layers instead of three, and it first performs pooling.

The main difficulty encountered when using the STN module with CNN is that there is a need to track the inadmissibility of retraining both networks. This leads to significant difficulties and instability of the learning process itself. At the same time, in order to avoid retraining of networks, it is sufficient to introduce a rather limited amount of augmented data into the training set (to a greater extent this concerns augmentation of image brightness). However, the above advantages of the implemented approach far outweigh its disadvantages.

4. Testing the developed system on real data

The developed system can carry out the detection of traffic signs and traffic lights (in this case, the system is trained to recognize the signs “pedestrian crossing” and “direction of movement”) both on static images and in video files.

To test the program, a video file was taken from the car recorder.

The video is fed to the program input, the processed video with detected objects is saved at the output. Screenshots with detected objects (highlighted with rectangles of various colors) are presented in Figure 1 and Figure 2.

![Figure 1](image1.png)

Figure 1. Traffic lights and signs “Pedestrian crossing” (fragment 1).
As you can see from the images, the application copes quite well with tracking signs “pedestrian crossing” (green rectangles in the photo). The signs are at different angles, relative to the DVR, which is shooting this video.

**Figure 2.** Traffic lights and signs “Pedestrian crossing” (fragment 2).

In this image, in addition to the “pedestrian crossing” sign, you can see traffic lights (olive-colored rectangle).

The following image (Figure 3) shows that the program also recognizes signs of the direction of movement (yellow rectangles).

**Figure 3.** Signs of the direction of movement (fragment 1).

Near the frames surrounding the detected objects, there are signatures pointing to the class of the detected object. The trafficlights class includes traffic lights, the zebra class is the pedestrian crossing sign, and the direction class is the traffic direction sign. Next to class signatures are percentages of object classification accuracy. The threshold value for detecting an object in an image is 90% or more.

Practical research allows us to talk about the level of detection of moving objects about 91%. But even with such a high accuracy of detecting objects, there are situations when an object in the image is classified incorrectly.

An example of insufficient quality detection of objects in the image is shown in Figure 4.

**Figure 4.** An example of an erroneous classification of an object.
Figure 4 clearly shows that one of the signs of the direction of movement was classified by the class “zebra” - that is, as the sign “Pedestrian crossing”. The minimization of such classification errors is achieved by increasing the size of the dataset on which the model is trained. Also, the dataset should include as many as possible photographs of the recognized object (different weather conditions, time of day, distance to the object, rotation of the object at different angles, relative to the camera lens, etc.).

Now we should focus on the effectiveness of training a neural network developed according to the method described in the previous section.

The accuracy of the classifier developed by the STN + IDSIA method is higher than that of IDSIA without the STN module by approximately 0.8% - 1.2%.

Already after the first 12 epochs, the accuracy of detecting moving objects in a video image turns out to be 90% (studies were conducted on a set of reference data) [18].

At the same time, the CNN neural network is still in the process of learning, however, it should be borne in mind that a double network of increased complexity is used on the original data set without expanding it with augmentations. Practice shows that with the addition of augmentation, it is possible to increase the recognition accuracy of moving objects to 95.2% in the first ten iterations (however, the time required for the learning process has significantly increased).

Figure 5 shows the learning outcomes of the models. This diagram displays the accuracy of the entire neural network on a set of reference data (the red curve is STN + IDSIA, the blue curve is IDSIA without the STN module).

![Figure 5. Model Learning Outcomes.](image)

Improving system performance can be achieved, for example, by applying the MSER method.

5. Conclusion
In the work, the purpose of which is to study the typification methods of moving objects based on a neural network, all the research tasks are solved. It turned out that the use of image classifications is so widespread today that it is difficult to imagine a sphere in which any algorithms are not implemented, of which the image classification process becomes a part.

The use of object tracking is advisable in the following tasks: recognition based on movements, human-computer interaction, navigation support for cars, for example, determining a route with obstacle detours based on video materials, monitoring traffic, that is, tracking traffic in real time with the purpose of a more correct direction of the flow of traffic, monitoring in an automated mode, that is, monitoring a specific scene of actions to identify unwanted events or aktivnosti suspicious nature [19-21].
The task of the developed system is to detect certain road signs (“Pedestrian crossing”, “Direction of movement”) and traffic lights on the video image of the road. To implement this system, Python was chosen. In developing the system, a CNN convolutional neural network with the IDSIA + STN module classifier was used.

An analysis of the results of testing the operation of the system on real video data allows us to talk about a sufficient level of detection efficiency of moving objects with a probability of up to 91%.

Acknowledgment
This work has been supported by the Russian Foundation for Basic Research, Grant No 19-29-06036.

References
[1] Al’-Mansur M, CHuaib I, Dzhafar A 2018 Giroskopiya i navigaciya 26-2(101) 43
[2] Buj V, Arhipenko I, Litvinov Y 2015 Nauchnyj al’manah 12-2(14)
[3] Buj V, Litvinov Y, Ho V 2016 Innovacionnaya nauka 4-3(16) 34
[4] Isaev A, Gazarov D, Evseev S 2016 Molodoy uchenyj 18 44
[5] Romanov A, Volkova M 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIconRus)
[6] Bujko A, Vinogradov A 2017 Programnye sistemy: teoriya i prilozheniya 4(35)
[7] Sikorskij O 2017 Novye informacionnye tekhnologii v avtomatizirovannyh sistemah 20 68
[8] Isaev A, Gazarov D, Evseev S 2016 Molodoy uchenyj 18 44
[9] CHujkov R, YUdin D 2017 Nauchnyj rezultat. Informacionnye tekhnologii 4 50
[10] SHashkov B, SHepelyev K 2017 Kombinirovannyj metod detektirovaniya i klassifikacii dvizhushchihsiya ob”ektov v sistemah videonablyudeniya (Moscow: Izdatel’stvo «Sputnik+») p 241
[11] An Tien Vo, Hai Son Tran, Hoang Le T 2017 9th International Conference on Knowledge and Systems Engineering (KSE)
[12] Chen K, Tripathi S, Hwang Y, Nguyen T 2016 International SoC Design Conference (ISOCC)
[13] Kulchandani Jaya S, Dangarwala Kruti J 2015 International Conference on Pervasive Computing (ICPC)
[14] Yashina M V, Tolmachev A A 2017 T-Comm. 8 46
[15] Lutc M 2014 Programmirovanie na Python (SPb.: Simvol-Plyus) p 992
[16] Guo T, Dong J, Henjian L, Gao Y 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)
[17] Tripathi Sh, Kumar R 2019 International Conference on Cloud Computing, Data Science & Engineering (Confluence)
[18] Rubanov M, Yakovlev D 2017 In the collection: Problems of automation and control in technical systems, collection of articles of the XXXII International scientific and technical conference 76
[19] Li J, Wang J, Mao J 2014 Proceedings of the 33rd Chinese Control Conference
[20] Naga Jyothi T, Vasavi S, Srinivasa Rao V 2015 IEEE International Advance Computing Conference (IACC)
[21] Chesnokov A, Gorodnichiev M, Gavriish K, Zhidkova M 2019 Systems of Signals Generating and Processing in the Field of on Board Communications, SOSG 2019 8706727