Robotic System Position Control Algorithm Based on Target Object Recognition

Pavel Slivnitsin, Andrey Bachurin and Leonid Mylnikov
Electrical Engeneering Faculty, Perm National Research Polytechnic University, Komsomolsky ave. 29, Perm, Russia
slivnitsin.pavel@gmail.com, bachurin@ellips.ru, leonid.mylnikov@pstu.ru

Keywords: Pattern Recognition, Image Recognition, Object Detection, Neural Network, Robotic System, Identification, Self-Positioning.

Abstract: Creation of robotic systems capable of manipulating objects of the real world is an actual problem allowing both raising labor efficiency and reducing traumatism risk for a person. In the paper, the task of large-node replacement of the outdoor lighting luminaires with the use of robotic system is considered. For this, the accompanying tasks connected with the object (luminaire) identification and positioning of robot gripper concerning object have been solved. The resulting algorithms allow us to solve tasks in conditions of varying visibility, different backgrounds, overlapping objects, if necessary, positioning on certain parts of the target object. They can be used to identify of any elongated shape objects after appropriate training of the neural network. These algorithms allow us to position the robot arm in such a way that it can take the necessary object with the help of the gripper. The practical significance of the solved problem is connected with the possibility of robotics systems practical application in the human environment and the creation of anthropomorphic robots.

1 INTRODUCTION

The development of automation tools and the algorithmic base so far has allowed to automate the majority of monotonous and similar operations, and to replace people on such operations and to increase labor productivity. Modern tasks facing automated and robotic systems [1] are related to the issues of independent collection and processing of information about the surrounding environment and its use for decision-making. This is largely due to the fact that the use of robotic and intelligent devices has left from the factory floor to the outside, where the location of the objects is not fixed. In addition, there are many modifications of objects that perform similar functions and various spatial combinations of objects.

The development of production systems within the framework of the Industry 4.0 concept will lead to the creation of virtual and intelligent automatic production that will have to operate not only according to predetermined algorithms, but also in emergency situations.

One such task is the outdoor lighting maintenance[2], namely the replacement of defective lighting equipment (luminaires) using a robotic system. This task is a good example of identification and self-positioning relative to a given object (a luminaire and a connector), by a robotic system for the replacement of outdoor lighting luminaires using large-node replacement technology [3]. To implement this technology it is possible to use the special connector [4] (Figure 1). Implementation of this technology is necessary for the device to be able to remove the defective luminaire and install a new one. When using this method in a large lighting network, the operation of replacing outdoor lighting luminaires becomes the same for each lighting installation and can be automatically performed using a robotic system (Figure 2).

To identify objects, the industry now uses technology based on the determination of their coordinates (device coordinates) based on binding to global coordinates, positioning based on predefined maps, technology based on the use of radio tags to identify objects, etc. Such technologies show good results in production, where all changes in space configuration are clearly tracked. When working in the environment, such technologies become extremely expensive due to the impossibility to equip all devices with tags and keep the environmental maps up to date. In such cases, image recognition technologies based on video and photo processing are used. Solutions for the recognition of some well-detectable objects (e.g. car license plates) have already been widely used. However, for self-positioning of technical devices, it is necessary to distinguish more complex objects under different conditions. The object recognition tasks are currently being solved as classification tasks, for which neural networks and machine learning methods have been used. Among neural networks, modifications of recurrent and convolutional neural networks (CNN) such as LSTM are widely used. Probabilistic neural networks (PNN) are used to identify objects with a variable shape and structure. SVM, decision trees, and Dalal-Triggs [5] and Viola-Jones methods [6] are used for individual images are the most frequently used among machine learning methods [7]. In some cases, methods of scale-invariant features transform (SIFT and SURF) are used. Preprocessing of images and models of multilayer perceptrons (MLP) using neural networks can be used to increase accuracy.

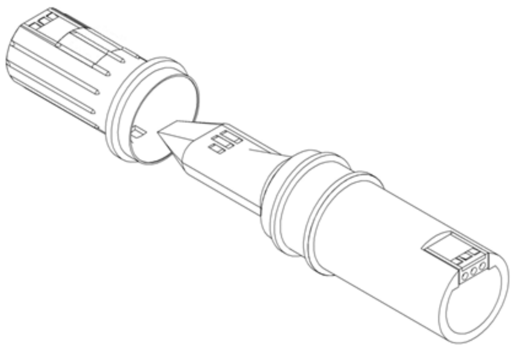

Figure 1: The connector scheme.

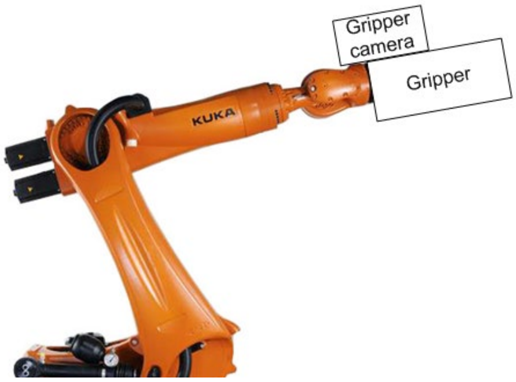

Figure 2: The structure of the robotic system.

The choice of a method to solve an applied task is associated with the speed of the identification and the accuracy of objects identification (Table 1), including the cases when there may be variability of shape and position, when the objects overlap, and when there are requirements to the speed of work and learning.

Table 1: Comparison of object recognition methods.

Method	Accuracy
SURF [8]	63%-90%
SIFT [8]	71%-91%, for some tasks 90%-99%
SVM [9]	77,69%-91,54%
Dalala-Triggs [5]	>90%
CNN [10]	90% - 97%

As we can see in the table, all modern methods for a certain class of tasks show good results. Therefore, the choice of the method will be based on the object peculiarities, the learning process and use of the object.

In the considered task, robotic system is an autonomous low performance device which operates in real time, that leads to high requirements to the speed of objects identification. Thus, training of model can occur independently, on a separate server or cluster, and the set of training sample is constantly replenished by the new images collected in the course of autonomous operation of the robotic system.

In such conditions, it is effective to use convolutional neural networks, which require the use of deep learning methods for their learning. The trained model can work effectively on low-power computing resources and gives high accuracy without using additional algorithms to increase accuracy [11].

2 METHODOLOGY

In outdoor lighting maintenance using a robotic system, the robot is delivered to the place of replacement using a lifting platform car. The car stops near the necessary support, and the robot identifies the luminaire. Then, it approaches the luminaire and the connector at a distance sufficient to perform the replacement procedure according to the algorithm shown in Figure 3.

The implementation of the above algorithm implies performing actions related to the object identification and robot positioning (rotation, shift and approaching operations).

Step 1: Target object = OUTDOOR LIGHTING LUMINAIRE.
Step 2: Photo / video shooting of the surrounding area.
Step 3: Target object identification.
Step 4: Perform a test move and determine the distance to the target object.
Step 5: Determine the Δ_1 step value for approach to the target object.
Step 6: Having approached the target object (step Δ_1), re-photograph the area and determine, the position relative to the target object, based on the analysis of two images.
Step 7: If the distance to the object is greater than Δ_2 then return to step 5, otherwise if the target object = OUTDOOR LIGHTING LUMINAIRE then the target object = LUMINAIRE CONNECTOR and go to step 5.
Step 8: Execution of the luminaire replacement algorithm.

Figure 3. The algorithm of the robotic system for the replacement of outdoor lighting luminaires.

2.1 Target Object Position Identification in the Image

To solve the task of object recognition, we will build a neural network on the basis of convolution neural network *Inception–v2* ([12] and [13]). The peculiarity of the network is in its structure, that uses complex structures representing modules as neurons (typical structures of neural networks): filtering and signal normalization modules, convolutional networks with the number of layers and branches from 2 to 5 and the number of neurons in each branch from 1 to 5, typical multilayer neural networks in which all neurons of the previous layer are connected to all neurons of the next layer with the number of layers from 1 to 5 and the number of neurons in each layer from 1 to 5. The use of such modules makes it possible to construct from them networks in which each layer is organized as a separate neural network with normalization of input and output signals, and the number of modules in each layer increases. To identify the luminaires, a network with 20 layers was used, 10 of which are Inception modules described in [13]. This structure in the learning process introduces specialization for constituent modules that begin to identify various special elements of an identifiable object. The structure of the selected neural network can be represented as shown in Table 2.

Table 2: Neural network architecture [13].

Layer type	Patch size/stride or remarks	Input size
Conv	3 × 3 / 2	299 × 299 × 3
Conv	3 × 3 / 1	149 × 149 × 32
Conv padded	3 × 3 / 1	147 × 147 × 32
Pool	3 × 3 / 2	147 × 147 × 64
Conv	3 × 3 / 1	73 × 73 × 64
Conv	3 × 3 / 2	71 × 71 × 80
Conv	3 × 3 / 1	35 × 35 × 192
3 × Inception module	As in figure 5 in [13]	35 × 35 × 228
5 × Inception module	As in figure 6 in [13]	17 × 17 × 768
2 × Inception module	As in figure 5 in [13]	8 × 8 × 1280
Pool	7 × 8	8 × 8 × 2048
Linear	Logits	1 × 1 × 2048
Softmax	Classifier	1 × 1 × 1000

The structure of the neural network affects the training process. Therefore, after selecting the structure of the neural network, it is necessary to optimize it. One of the main reasons why this is necessary is that when training the network, it is possible to move incorrect information from the end of the network to all the weights inside. In this case, if to change the parameters of the input layer is a simple task, then to get access to the parameters of the layers behind the first one is not a simple task. It is possible to write formulae for updating the weights within the network. However, due to the fact that each neuron depends on the other with which it has a connection, various new problems may appear, such as: 1) sticking at local minima; 2) infrequent updating of rare features, which negatively affects the capabilities of the generalizing network rule, but on the contrary, great emphasis on rare features leads to the network retraining.

When implementing a neural network, neurons with the ReLU activation function should be used to identify objects. Calculation of the sigmoid and hyperbolic tangent requires more resource-intensive operations, such as exponentiation, while the ReLU can be implemented using a simple threshold conversion of the activation matrix at zero. In addition, the ReLU is not saturated.

The use of ReLU increases the learning speed significantly. That is caused by linear character and absence of saturation of this function.

Special attention should be paid to the choice of optimizer. The speed of the learning mechanism and the stability of the result will depend on this (comparison of the work of several optimizers is shown in Figure 4).

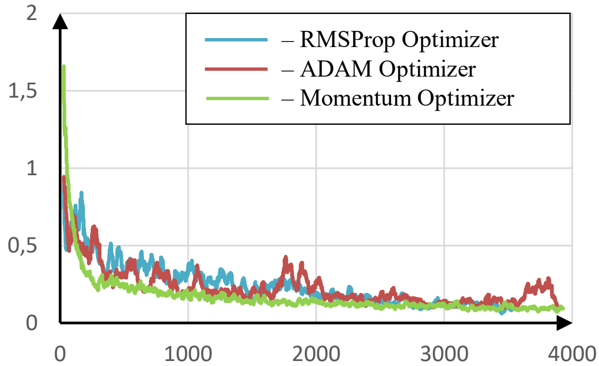

Figure 4: Graphs of changes in the value of the loss function when using various optimizers.

2.2 Rotation Operation

It is not enough to identify the object in the image, it is also necessary to define the robot position relative to the target object. First of all, the task of gripper rotation is solved. The solution of this task is to determine the angle of rotation for the correct orientation relative to the luminaire (target object) (Figure 5).

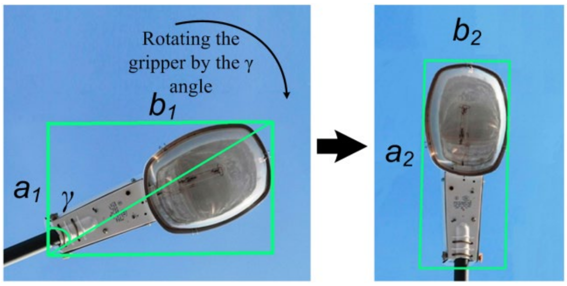

Figure 5: Determination of the rotation angle.

The solution of this task is performed using formulae (1) and (2).

$$\frac{b_1}{a_1} \neq \frac{b_2}{a_2}; \frac{b_2}{a_2} = \frac{B}{A}; \qquad (1)$$

$$\gamma = \arctan\left(\frac{b_2}{a_2}\right); \qquad (2)$$

where a_1, b_1, a_2, b_2 are the frame dimensions of a detected object (luminaire) in the image in pixels; A, B are the actual luminaire size characteristics in meters; γ is the manipulator gripper rotation angle.

2.3 Shift Operation

After the rotation operation and before approaching to the object, it is necessary center the object relative to the center of the image (Figure 6). This is necessary so that there is no gripper shift relative to the object when approaching it.

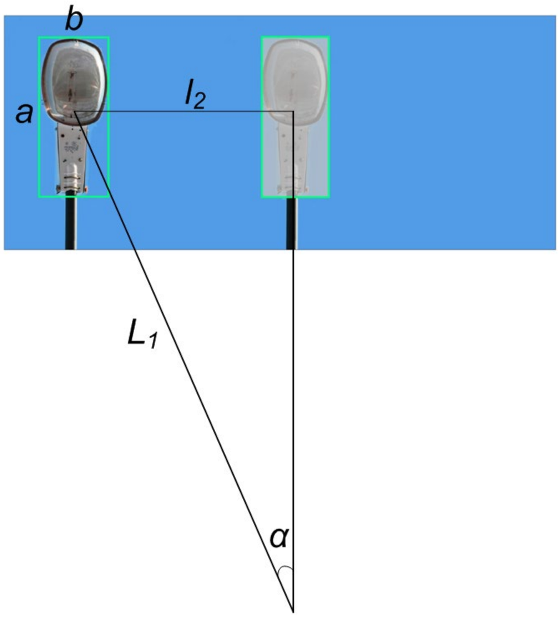

Figure 6: Determination of the angle of robot gripper.

To do this, it is necessary to determine the angle through which the robot should be rotated; this angle is determined by the Formula 3.

$$\alpha = \arcsin\left(\frac{L_2}{L_1}\right) = \arcsin\left(\frac{l_2 A}{L_1 a}\right); \qquad (3)$$

where α is the angle through which the robot must be rotated; l_2 is the distance from the object to the center of the image in pixels; a, b are the frame dimensions of a detected object (luminaire) in the image in pixels; A, B are the actual luminaire size characteristics in meters; L_2 is the distance from the object to the center of the image converted into meters; L_1 is a measured distance from the gripper camera to the object in meters (approaching operation).

2.4 Approaching Operation

The next operation to be performed is to define the value of the next step to approach to the object. For this, it is necessary to define the distance to the object (Figure 6).

Considering that the object can be photographed at each step of the algorithm (Figure 3), it is possible to compare the last two images, namely, to compare the geometric dimensions of the detected object and to estimate the distance to the object. For this, it is necessary to solve the system of equations (4) (Figure 7).

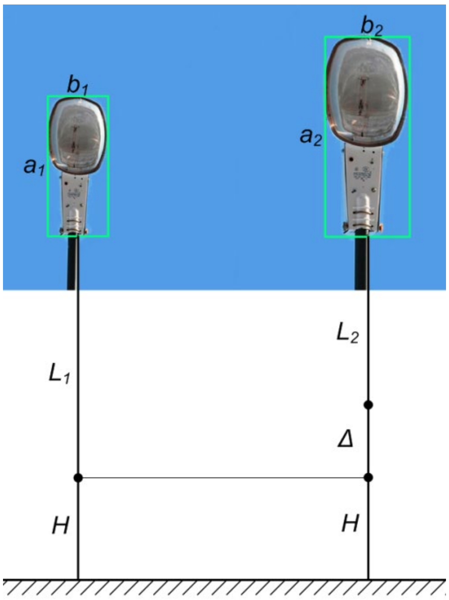

Figure 7: Determining the distance to the object.

$$\begin{cases} \frac{L_1}{L_2} = \frac{L_2+\Delta}{L_2} = k\frac{a_2}{a_1} \\ \frac{L_1}{L_2} = \frac{L_2+\Delta}{L_2} = k\frac{b_2}{b_1} \end{cases}; \quad (4)$$

where H is the camera height in meters; Δ is the distance travelled towards the object in meters; a_1, b_1, a_2, b_2 are the frame dimensions of the detected object (luminaire) in the image in pixels; k is the pixel-to-meter conversion factor (due to the use of two images, the values obtained from the system (4) are independent of the k value); L_1, L_2 are the required distances from the camera to objects in meters.

3 IMPLEMENTATION

Implementation of an algorithm to identify outdoor lighting luminaires requires neural network training.

To train a neural network, it is necessary to form a training sample of images with luminaires labelled on them. To label luminaires in the image, a rectangular area is selected, which coordinates are recorded in an XML file (Figure 8):


```
<annotation>
    <folder>train</folder>

 <filename>IMG_20191101_164622.jpg</fil
ename>

 <path>C:\tensorflow1.1\models\research
\object_detection\images\train\IMG_2019
1101_164622.jpg</path>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>195</width>
        <height>260</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>luminaire</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>30</xmin>
            <ymin>44</ymin>
            <xmax>161</xmax>
            <ymax>144</ymax>
        </bndbox>
    </object>
</annotation>
```


To create labels, we use some tools (for e.g. *LabelImg*).

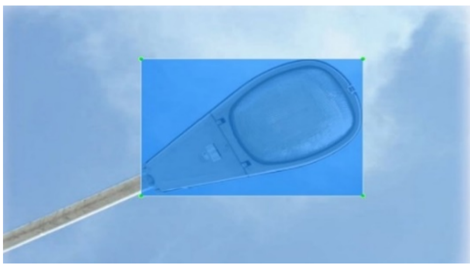

Figure 8: An example of labelling an object in an image.

After all images have been labelled, the coordinates of all selected areas are recorded to a ".csv" file, which contains the image names and coordinates of the selected areas in them.

For correct training of the neural network, it is necessary to use images of different types of luminaires from different angles, in different conditions in which we can observe them (examples in Figure 9).

The initial dataset for training of the neural network consisted of 250 images.

To test the object identification in the image, the convolutional neural network, the *TensorFlow* library to work with neural networks and its add-on for object detection in images (*Object Detection API*) were used.

Having prepared a set of images for training, select the previously prepared model [14]. *TensorFlow Object Detection API* provides a number of pre-trained models. The use of a pre-trained model significantly reduces training time. The model *"faster_rcnn_inception_v2_coco"* is selected for testing.

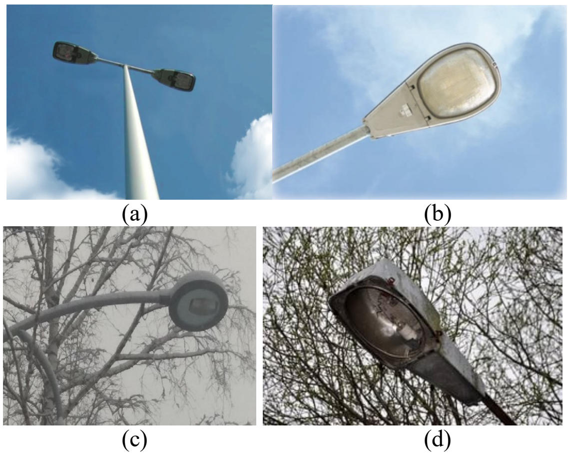

Figure 9: Examples of images from the training set.

As a result of the neural network training, rotation method implementation and approach-to-target method application, the algorithm shown in Figure 3 is implemented as a sequence of steps, an example of which is shown in Figure 10.

4 DISCUSSION

The task considered in this paper is an element of a complex solution consisting of two operations: the removing operation and installing operation of an outdoor lighting luminaire. Each of these operations consists of many subtasks, some of which can be invariant. For example, if we consider the luminaire removing operation, it can be decomposed into a number of algorithmic subtasks, as shown in Figure 11.

The solution of some tasks can be invariant. For example, the task of approaching to the target object in practice, using the proposed solution, can be performed in different ways. They will depend on the initial position, conditions affecting the quality of the images (for example, weather conditions and overlapping), and this does not take into account obstacles that may be in the way of the robot (for example, tree branches). Identification and grasping of objects by robotic systems are repeated many times. It means that the accumulated experience (photos and motion trajectories leading to successful results and failures) can be used for the training of the object detection model and improvement of the trajectory choice model based on machine learning methods or using intelligent algorithms.

Other tasks, which need to be solved for the luminaire removing operation, also have the property of invariance and the possibility of the algorithm modernization.

In this regard, at the moment, one can observe a variety of ways to solve these tasks, that combine attempts to create universal algorithms, the use of which will allow them to be used without relating to a particular object. One of the methods described in the literature when using neural networks for task recognition is the search for universal structures of such networks and use of Siamese networks [15].

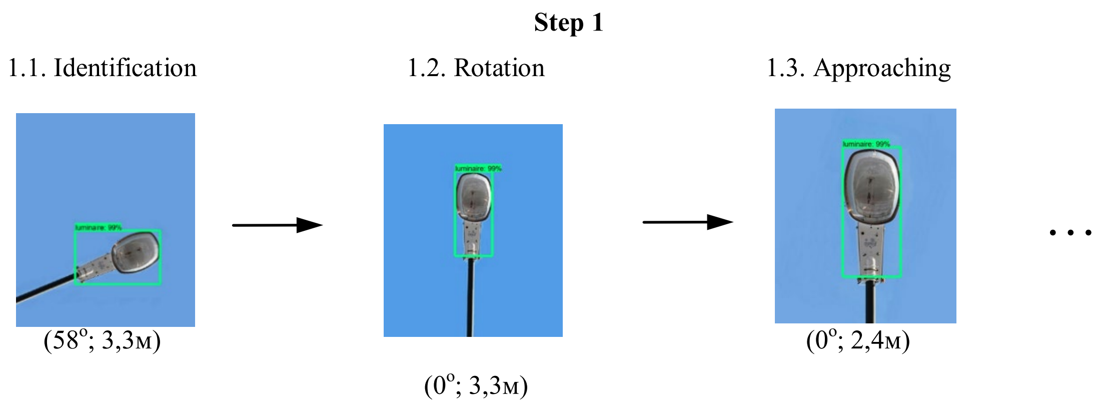

Figure 10: Identification and determination of the distance to the luminaire.

The subtasks shown in Figure 11 are not unique. They constantly arise in the control and creation of robotic systems. Therefore, today there is a great variety of task solutions to find an object and grasp it, approach the manipulator to the target object and the gripping point. In literature, there are algorithms using stereo vision [16]; identification of objects based on the choice of primitives that they consist of [17]; search for objects in the image and their classification using neural networks [18]; performing operations related to inserting one object into another (insertion of wires in the terminals, in our case installation of a new luminaires) [19]; tasks related to the movement of the gripper with an object that are considered as tasks of bending around obstacles on the basis of predetermined trajectories [20].

The presence of a large number of algorithms indicates the relevance of solving these algorithmic subtasks. The solution of tasks is associated with compromises between quality and complexity of the hardware, complexity of the algorithms and the speed of their work. Today, there are solutions that demonstrate good results in laboratory conditions. However, they do not always work well in real conditions due to the factors of the environment in which they operate.

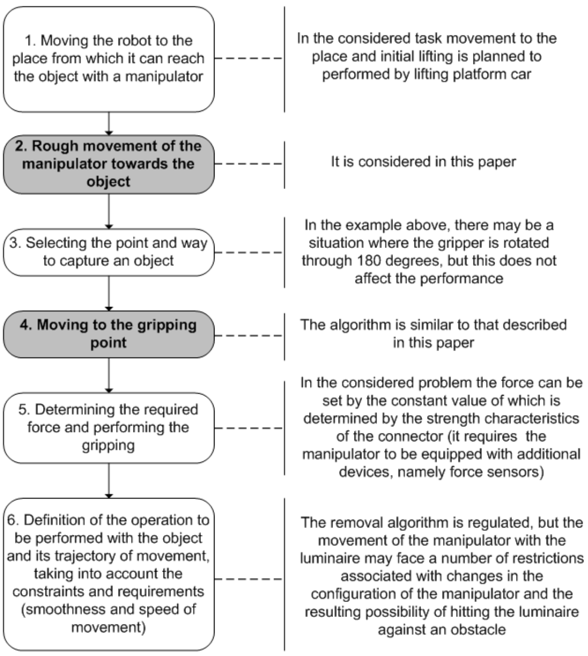

Figure 11: The scheme of dividing the luminaire removing task into algorithmic subtasks (the tasks considered in the paper are highlighted in gray).

5 CONCLUSIONS

The key task that was solved in the study described in the paper is the task of identifying objects. The use of neural networks is an effective tool that does not depend on a particular subject area.

The resulting model, trained on a dataset of 250 images, identifies the outdoor lighting luminaires in the images with high accuracy, but for some objects, much more images may be required to improve accuracy.

The approach considered in the paper relates to the subject area only through the object, the recognition of which takes place. In this regard, it can be considered universal and used to solve other applied tasks associated with the fact that an object must be taken and certain actions must be performed with it. Performing such manipulations is especially important when implementing humanoid robots capable of performing actions related to household and industrial activities. [21]. The creation of such robots is one of the priorities of robotics and is required to implement a complete Turing test [22].

REFERENCES

[1] A. Dudarev, “The Problem Sensitization Robotic Complex Drilling and Milling of Sandwich Shells of Polymer Composites,” Proc. 4Th Int. Conf. Appl. Innov., vol. 4, no. March, 2016, pp. 15-19.

[2] N. Pavlov, A. Bachurin, and E. Siemens, “Analysis of Outdoor Lighting Control Systems and Devices for the Creation of Outdoor Lighting Automatic Control System Using the Traffic Flow Value,” Proc. Int. Conf. Appl. Innov. IT, no. March, 2017, pp. 95-100.

[3] P. Slivnitsin and A. Bachurin, “A modern way of outdoor lighting maintenance,” International Conference on Innovation Energy, 2019, J. Phys.: Conf. Ser. 1415 012010.

[4] А.А. Бачурин и П.А. Сливницин, “Соединительное устройство для монтажа и подключения светильника наружного освещения,” 2695631, 2019.

[5] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” Proc. - 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, CVPR 2005, vol. I, no. 16, 2005, pp. 886-893.

[6] P. Viola and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” in Proceedings IEEE Conf. on Computer Vision and Pattern Recognition, pp. 511-518, 2001.

[7] L. Mylnikov, B. Krause, M. Kütz, K. Bade and I. Shmidt, Intelligent data analysis in the management of production systems (approaches and methods), Shaker Verlag {GmbH}.

[8] V. Renò et al., “A SIFT-based software system for the photo-identification of the Risso’s dolphin,” Ecol. Inform., vol. 50, pp. 95-101, January 2019.

[9] N. Kumar et al., “Leafsnap: A computer vision system for automatic plant species identification,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7573 LNCS, no. PART 2, 2012, pp. 502-516.

[10] H. Zhou, C. Yan and H. Huang, “Tree species identification based on convolutional neural networks,” in Proceedings - 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2016, vol. 2, 2016, pp. 103-106.

[11] F. Al-Azzo, A. M. Taqi and M. Milanova, “Human related-health actions detection using Android Camera based on TensorFlow Object Detection API,” Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 10, 2018, pp. 9-23.

[12] C. Szegedy et al., “Going deeper with convolutions,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 07-12-June, pp. 1-9, June 2015.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 2818-2826, December 2016.

[14] “Tensorflow detection model zoo.” [Online]. Available: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md.

[15] S. Ravichandiran, Hands-On Meta Learning with Python: Meta learning using one-shot learning, MAML, Reptile, and Meta-SGD with TensorFlow. Packt Publishing Ltd.

[16] F. Chen, M. Selvaggio and D.G. Caldwell, “Dexterous Grasping by Manipulability Selection for Mobile Manipulator with Visual Guidance,” IEEE Trans. Ind. Informatics, vol. 15, no. 2, 2019, pp. 1202-1210.

[17] H. Dong, E. Asadi, G. Sun, D. K. Prasad and I.M. Chen, “Real-Time Robotic Manipulation of Cylindrical Objects in Dynamic Scenarios Through Elliptic Shape Primitives,” IEEE Trans. Robot., vol. 35, no. 1, 2019, pp. 95-113.

[18] C.H. Corbato, M. Bharatheesha, J. Van Egmond, J. Ju and M. Wisse, “Integrating different levels of automation: Lessons from winning the amazon robotics challenge 2016,” IEEE Trans. Ind. Informatics, vol. 14, no. 11, 2018, pp. 4916-4926.

[19] D. De Gregorio, R. Zanella, G. Palli, S. Pirozzi and C. Melchiorri, “Integration of robotic vision and tactile sensing for wire-terminal insertion tasks,” IEEE Trans. Autom. Sci. Eng., vol. 16, no. 2, 2019, pp. 585-598.

[20] Z. Cao, N. Gu, J. Jiao, S. Nahavandi, C. Zhou and M. Tan, “A Novel Geometric Transportation Approach for Multiple Mobile Manipulators in Unknown Environments,” IEEE Syst. J., vol. 12, no. 2, 2018, pp. 1447-1455.

[21] A. Billard and D. Kragic, “Trends and challenges in robot manipulation,” Science (80), vol. 364, no. 6446, p. eaat8414, June 2019.

[22] A.M. Turing, “Computing machinery and intelligence,” Mind, vol. LIX, no. 236, 1950, pp. 433-460.