Object Recognition Based on Three-Dimensional Computer Graphics

Nikolay Lashchik(✉)

MIREA – Russian Technological University, Vernadsky Avenue 78, 119454 Moscow, Russia
lkolya97@gmail.com

Abstract. This paper considers the issue of training convolutional neural networks based on the representation of a data set in the form of three-dimensional computer graphics. The principle of operation of convolutional neural networks is to train and recognize categories of objects using orthogonal projections of a three-dimensional object. In the recognition process, difficulties arise in mirroring the object, which are solved by using the Pearson correlation coefficient. To use a convolutional neural network, a comparative analysis of the architectures of existing effective image recognition solutions for recording visual information was carried out. For the correct operation of the trained neural network, three-dimensional graphics objects were modeled. After obtaining the desired three-dimensional model, a black-and-white image was obtained, which was used to obtain the contours of the white spots. It is on such black-and-white drawings that the object recognition program is trained. The success of the neural network has been proven experimentally. Thus, it is possible to recognize real objects based on convolutional neural networks trained in virtual space.

Keywords: Three-Dimensional Object · Object Category Recognition · Object Mirroring · Convolutional Neural Networks · Machine Learning

1 Introduction

Object recognition is one of the most important cognitive functions of the human brain. Firstly, the recognition process begins with the process of perceiving an object. This process of perception is carried out with the help of the organs of perception, chiefly vision. Contemplation as a way of cognitive activity helps a person read information about the distinctive features of the observed objects [1].

Regarding artificial intelligence technologies, the problem of object recognition has been successfully solved by computer vision based on convolutional neural networks. With their help, two-dimensional graphical information can be manipulated and transformed. However, there are situations when it is impossible to provide a training sample in a timely manner. For example, a person has the ability to easily adapt to different situations and environments by observing new categories of objects. On the contrary, to transfer an automatic device to a new environment, it is often necessary to completely change the new knowledge base. It is obvious that many important issues in the field of

V. Jordan et al. (Eds.): HPCST 2023, CCIS 1986, pp. 132–142, 2024.
https://doi.org/10.1007/978-3-031-51057-1_10

computer vision have been successfully solved, but there are tasks that require improvement in their solutions. This paper discusses the issues of training a convolutional neural network based on three-dimensional computer graphics.

2 Problem Statement

Object recognition using very limited training data is crucial for many computer vision applications. This task becomes even more difficult when the system needs to learn about new categories of objects from very few examples. It seems very difficult to use convolutional neural networks for such tasks, since they require a lot of data and are prone to catastrophic forgetfulness. However, this problem can be solved on the basis of the instance-based learning and recognition approach, which considers the study of categories as a process of studying instances of a category, i.e., a category is represented only by a set of known instances, C ← {O_1,…,O_n}, where is a representation of the type of object based on combining layers [2].

Instance-based learning, also known as memory-based learning, is a basic approach to evaluating the representation of objects. The advantage of the instance-based approach compared to other machine learning methods is the ability to quickly adapt the object category model to a previously invisible instance by saving a new instance or deleting an old one. This approach to learning is able to recognize objects using a few exponential instances while maintaining too many redundant instances, which leads to large memory consumption and slows down the recognition speed [3]. Therefore, every time a new request appears, its previously saved data is checked, and a value for the new instance is assigned to the target object.

3 Related Work

At the moment, the following methods are used to solve the problem of insufficient training data: the first and most common method, which can help with the lack of training data, is completely avoiding test samples, but this option is the least effective method of solving this problem. Its main advantages are ease of execution and time savings. However, the consequences of such a decision are difficult to predict. If the models combining input and output variables are simple enough, the resulting neural network model may be suitable enough, but verifiability will only appear at the stage of practical work, when the cost of the solution resulting from the interpretation of the results will increase disproportionately. For complex or poorly understood models, it is better to immediately abandon this approach.

The second method is to apply the theory and methods of fuzzy logic and fuzzy sets. The idea is to create "if-then" rules to determine the relationships between different input parameters, which allows you to solve the problem of inconsistency of data and their insufficient quantity. Using these rules and methods, new examples are created that help increase the amount of available data for training. As a minus of this approach, it can be noted that the use of fuzzy logic methods can lead to a decrease in the accuracy and reliability of the model since they are based on fuzzy "if-then" rules that may be less accurate and objective than traditional machine learning methods. In addition, the creation and processing of such rules can be a complex and time-consuming process that requires deep knowledge in the fields of fuzzy logic and mathematical modeling.

The third approach offers a solution to the problem of a lack of training data by dividing input and output variables into smaller groups. Then simpler artificial neural networks - single-layer perceptrons—are applied to each of these groups. These perceptrons can be trained using the available data volumes. After learning, the perceptrons are combined into a single structure called the perceptron complex. The disadvantage of this approach may be a decrease in the accuracy of the model, since the separation of variables into groups can lead to a loss of information and a decrease in the detail of the model. In addition, the process of dividing variables into groups and choosing the optimal number of groups can be quite complicated and require a lot of experiments and testing.

4 Proposed Solution

Recently, considerable attention from the perspective of computer vision has been paid to the in-depth study approach, which can be divided into three methods depending on their input data: data based on volume, where the object is represented as a three-dimensional voxel grid, which is defined as input data for a convolutional neural network; data based on form, where the object is represented by a set of two-dimensional images by projecting object points onto a plane; and data based on a set of points where the object is represented behind a previously marked three-dimensional image. Among these methods, the most effective in the recognition of objects were the methods based on form; they allowed the researchers to achieve the best recognition results to date. At the same time, this representation greatly simplifies the study of the object in the environment of creating three-dimensional computer graphics “Blender”.

Thus, the point cloud of a virtual object is represented as a set of points p_i, where $i \in \{1, \ldots n\}$. . Then each point can be represented by x, y, and z coordinates. Based on the analysis of eigenvectors, three main axes of the object are constructed, and the center of the object is determined using the capabilities of the Blender software. On this basis, orthogonal projections can be implemented to create object views. Up to six types of projections can be implemented, but three are enough to represent a three-dimensional object, including a front, top and right view (Fig. 1).

It should be noted that the direction of the eigenvectors is not unique, orthogonal projections can be mirrored. The Pearson correlation coefficient is used to eliminate ambiguity. Then the projected point $\rho = (\alpha, \beta) \in \mathbb{R}^2$, where α is the distance perpendicular to the horizontal axis, and β is the distance perpendicular to the vertical axis. The Pearson correlation coefficient r is calculated for the direction of the X axis by the formula:

$$r_x = \frac{\sum(\alpha_i\beta_i - \overline{\alpha}\overline{\beta})}{\sqrt{\sum(\alpha_i - \overline{\alpha})^2}\sqrt{\sum\left(\beta_i - \overline{\beta}\right)^2}}, \quad (1)$$

where $\overline{\alpha}$ и $\overline{\beta}$ - average value α and β. According to the formula, the properties of the correlation coefficient r are determined: the Pearson correlation r varies in the range

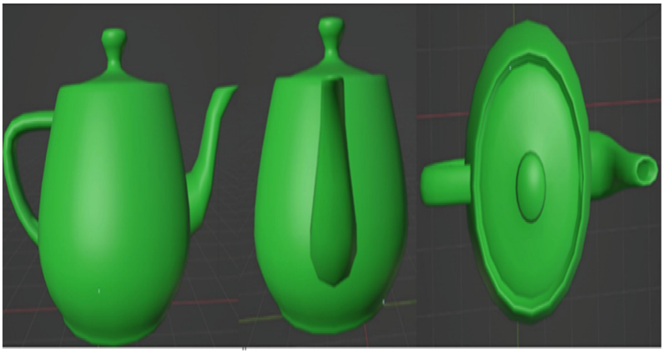


Fig. 1. Example of an object and its orthogonal projections.

from -1 to $+ 1$; the value of the Pearson correlation r indicates how close the points are to a straight line.

In particular, if Pearson correlation r = + 1 there is an absolute positive correlation, and if Pearson correlation r = -1 there is an absolute negative correlation. Similarly, the indicator is calculated r_y for the Y axis using a plane YoZ. Further, the sign of the axes is defined as $s = r_x r_y$, where s can be either positive or negative. In the case of a negative value of s, the three projections should be mirrored, while in the case of a positive value of s, they should not.

When the dissimilarity of the new object with all previously known categories exceeds the threshold value, the automatic device concludes that this object does not belong to known categories and thus initializes a new category marked as "category_m + 1", where m is the number of currently known categories. Moreover, in the case of a similarity measure, the difference between two objects can be calculated using various distance functions. For example, Pearson chi-squared is used to estimate the similarity of two instances.

When processing each image, it is necessary to implement the detection of the contours of objects (Fig. 2).

As a result, an image will be obtained (Fig. 2) on which all the objects you are looking for will be white spots and the background will be completely black. Next, you can use the built-in OpenCV library's findContours function to obtain the contours of white spots representing the desired objects.

A convolutional neural network can be considered a special kind of artificial neural network adapted for effective pattern recognition in an image. It implements some features of the visual cortex of the brain, in which two types of cells were discovered: simple, reacting to straight lines at different angles, and complex, whose reaction is associated with the activation of a certain set of simple cells. There are many implementations of neural networks of this type for classifying objects in images and image recognition.

Among the tested neural networks, the models of the YOLOv3 group look the most attractive, namely YOLOv3–320, YOLOv3–416 because of their high performance (> 30 FPS) and accuracy that is not lower than the average of the others (> 50%). The model is based on the Darknet framework. Despite the different names, in fact it is the

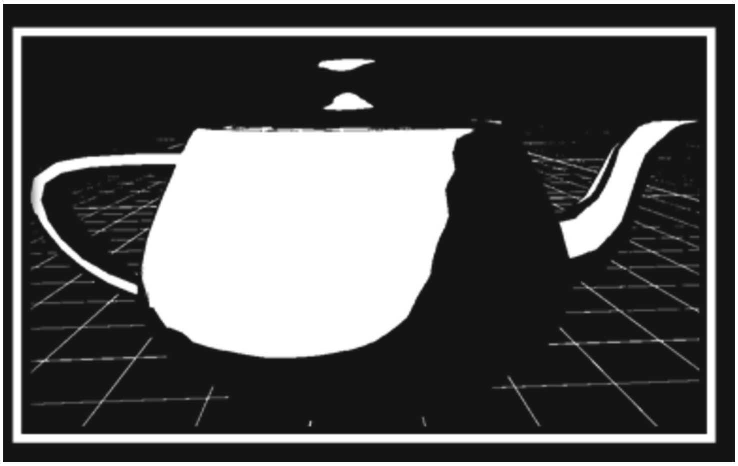


Fig. 2. The result of image processing.

same model, and the addition of 320 and 416 in the name denotes the width of the image coming to the input.

A 448 × 448 image is fed to the network input, which, in case of inconsistency, will be stretched or compressed to this size. The network consists of 106 layers and uses only 4 types of layers and 2 types of operations on layers (Fig. 3).

The first type is a convolutional layer. It creates a feature map based on the input data, in which the main patterns found in the image are highlighted. The second layer is the routing layer, which is necessary to determine the two layers, the result of which must be taken in order to perform the merge operation. The third type is the superselection layer. It increases the output dimension due to the feature map obtained from the convolutional layer. This is necessary for the correct execution of the merge operation. The fourth type is the detection layer. It is used to determine the location of the framing windows. The union operation concatenates feature maps from two layers, and the addition operation sums feature maps element by element.

Before training, it is necessary to configure the network configuration. To do this, it is necessary to set the number of classes equal to the original on the detection layers and, on the convolutional layers preceding them, change the dimension of the feature map to a value equal to the number of classes [4]. At the same time, at each epoch, the original images were subjected to various modifications, for example, turning the image by a small angle, changing the color scale, etc. In order to achieve better recognition quality, the weights of an already trained neural network were used on a set obtained in the software for creating three-dimensional computer graphics “Blender”. YOLOv3 uses the sum of root-mean-square errors as an accuracy metric

$$E_{MSE} = \frac{1}{n}\sum\nolimits_{j=1}^{n}\left(d_j - y_j\right)^2. \tag{2}$$

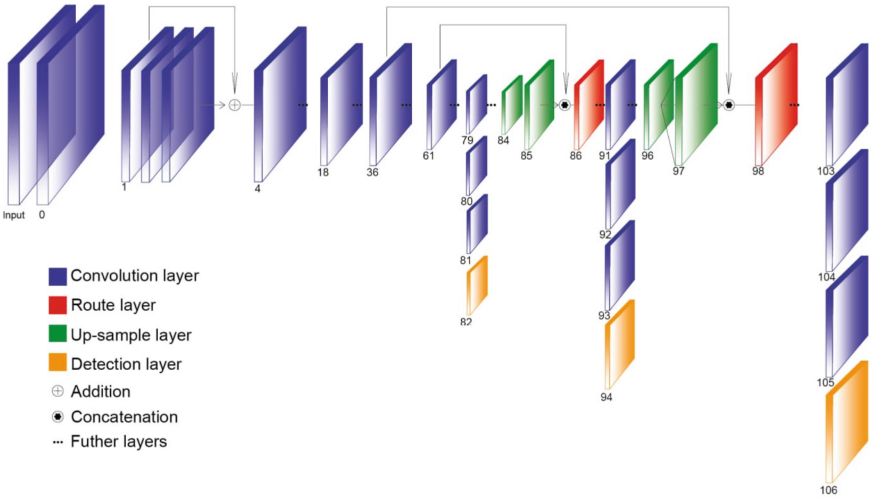


Fig. 3. YOLOv3 network architecture.

In machine learning theory, learning error is understood as the difference between the desired (target) d_j and the actual y_j output of the model using examples of the training set with the number of observations n.

When developing an automatic system based on computer vision, it is not enough to implement only the search for objects in the frame. Since a continuous stream of real-time video is received at the input, it is also necessary to learn how to programmatically track the movement of found objects. To do this, object trackers come to the rescue. An object tracker is an algorithm that determines the location of a moving object (or several objects) over time. Tracking a moving object can be a time-consuming process due to the amount of data contained in the recording of visual information (video). The complexity is further compounded by the possible need to use object recognition techniques for tracking, which is itself a difficult task [5].

The most popular when the movement is predictable and small is the MedianFlow tracker. The principle of operation of this tracker is as follows [6]:

1. The algorithm splits the image of the tracked object into small fragments
2. For each fragment, a new position is found on the next frame by calculating the optical flow between the two images.
3. To estimate the new position of the object, the tracker calculates the median of the displacement vectors of all image fragments.
4. To estimate the change in the size of the tracked object, the tracker calculates the median of the distances between all fragments of the image.

The tracker tracks the object both forward and backward in time and measures the discrepancies between these two trajectories. Next, the tracker selects the trajectory with the minimum error (difference) between the two frames. Unlike other trackers that continue to work and try to track an object even when it is either “lost” by the tracker or disappears from the frame, this tracker is able to determine when tracking failed. This is one of the main advantages of this tracker.

An increase in the objects being determined by an automatic system can allow identifying various situations at the level of visual-effective and visual-imaginative thinking. In order to move from the usual recognition of objects based on perception, in our case visual, to recognition by finding various patterns, it is necessary to perform the complex and extremely time-consuming work of identifying various patterns characteristic of a particular object.

At the same time, data sets for training a convolutional neural network can be taken from virtual reality, and testing is carried out in a real environment using visual information recording. This approach, according to the authors, will create an automatic device that allows not only to recognize real objects but also to determine the situation, which is also caused by virtual exposure.

5 Proof of Work

After the implementation of the architecture and algorithm of the convolutional neural network, it becomes necessary to verify the correctness of the implemented program.

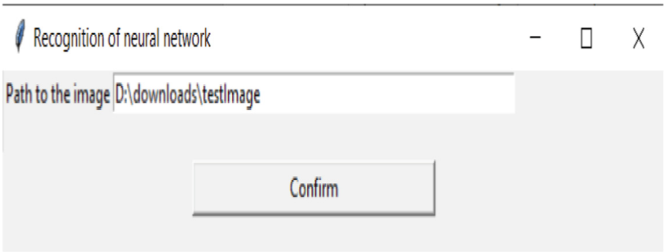
First and foremost, you need to make sure that the lack of training data is really a critical problem when training neural networks. To prove this statement, the neural network was trained to recognize a teapot on a small amount of data. In this particular experiment, 10 photos of a teapot were taken, although it is believed that the minimum number should be at least 100.

After training the neural network, photos of dummies from a different set, different from the one on which the network was trained, were transmitted. Out of a hundred photos, a neural network trained on an insufficient amount of test data correctly identified the subject only 20 times, which, as it is easy to calculate, is only 20% of successful test results.

After receiving the result with which the work of the program proposed in this article will be compared, it is necessary to implement a simple and intuitive interface that would show the work of the program. It is necessary to conduct tests not only to prove the correct operation of the neural network, but also for cases where incorrect data for analysis and objects unknown to the program were provided. For these purposes, a program was implemented to check the definition of the object depicted in the photo. This program has a simple window interface (Fig. 4), in which you need to enter the path to the file with the image of the object to be recognized. After clicking on the confirmation button, the program should output a message with the name of the object that the neural network recognized in the photo.

If you enter the wrong path to the file, a message is displayed stating that the wrong path to the image is specified. This reaction occurs when you enter a path where no file exists or if a file with an extension other than the extension that is an image file is located at the specified address.

If an image is submitted to the input of the program that does not depict a teapot, then, as expected, the program cannot recognize this object and issues a message about it.



D:\downloads\testImage

Fig. 4. Interface of the window application for checking the operation of the program.

After all possible variants of the unsuccessful outcome of the program have been checked, you can test the reaction of the neural network to the picture showing the teapot.

For the recognition test, a photo of a real teapot was taken (Fig. 5), which looks like a three-dimensional model of a teapot on which the neural network was trained.

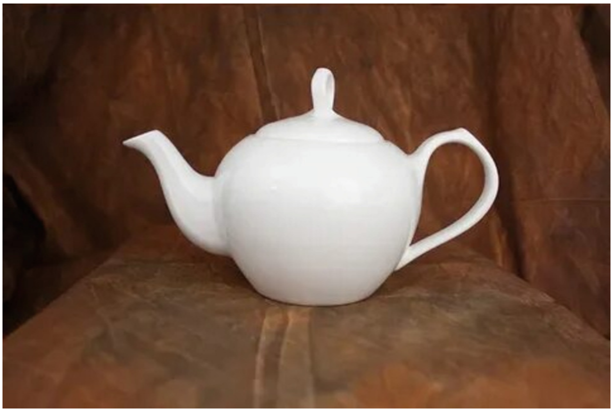


Fig. 5. The photo on which the first test of the program was carried out.

After launching the program, it is clear that the neural network correctly recognized that the teapot is located on the entered image.

After the initial successful test, another image of the teapot was selected (Fig. 6a), which is slightly different from the first test image and the model on which the program was trained. The photo taken differs from the first one in the presence of gold edging on the upper parts of the teapot and its lid.

After launching the program, a response was received that the neural network recognized a teapot in the input image. This test shows that the presence of some additional colors on the tested image does not affect the correctness of the program output. For the third test, a photo of a teapot of a different color with patterns applied to it was selected (Fig. 6b).

The program responded to the entered photo by recognizing a teapot in it. The test can be considered successful. He shows that a trained neural network can recognize teapots not only in white but also with slight differences from the images that were

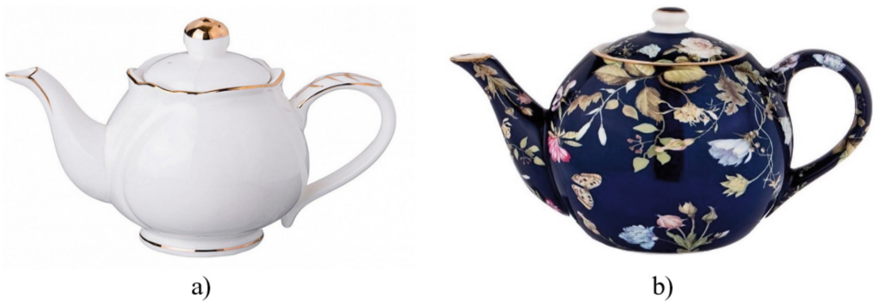


Fig. 6. Sample objects in the images used for the second and third tests: a) teapot that is quite similar in design to those used in training set; b) teapot that slightly varies from those used in training set (features complex color patterns).

used in the training of artificial intelligence. The developed program is able to recognize objects of different colors with various additional images on them.

The fourth test was conducted on the basis of a photo of a teapot, which is very different from the object of the training (Fig. 7a). In addition to the presence of a handle, it is also distinguished by its shape and a shortened spout. Despite the noticeable differences in the test object, the program was able to recognize a teapot in it.

A mug photo was selected for the final demonstration test (Fig. 7b). This choice was made due to the fact that the mug is close to the teapot in meaning but nevertheless differs in its functionality and representation.

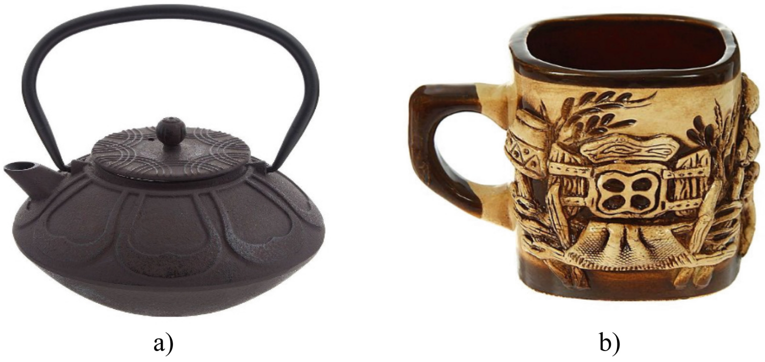


Fig. 7. Samples of objects in the images used for the fourth and fifth tests: a) teapot that is significantly different from the training set; b) fancy cup with partial visual resemblance of a teapot.

On the entered photo, the program gave an error that it could not recognize the object that is depicted. This test shows that a trained neural network correctly recognizes only the objects on which it has been trained. Thus, it is possible to perceive this test as passing positively.

Table 1. Results of quantitative evaluation.

Object type	Number of tests	Accuracy
Teapot similar to training set (as in Fig. 6a)	100	93
Teapot that varies slightly from training set (as in Fig. 6b)	100	87
Teapot different from the training set (as in Fig. 7b)	100	79
Different object of houseware (as in Fig. 7a)	100	99

Next, a larger-scale testing of the neural network was carried out, in which images of teapots were transmitted to the program for recognition, as similar as possible to the objects on which it was trained, slightly different from them, and very different (Table 1). 100 photos were selected for each category of test objects to be able to approximate the percentage of correct operation of the program. Each group was selected according to the same criteria, namely, image quality, that is, photos were selected in which objects occupying most of the picture are clearly visible. It should be clarified that the dataset does not pretend to be the result of extensive and detailed testing but is simply used to approximate the results of the program since there is no ready-made solution for such testing.

Testing was also carried out, in which other kitchen implements were transferred for which the program was not trained. In this case, the successfully passed test is the one in which the neural network did not recognize the teapot.

After the tests of the program, it can be seen that the developed and trained neural network correctly recognizes the teapot in the photos, which means that this approach to training an artificial network on three-dimensional image models is effective.

Naturally, if a teapot is depicted in the photo but is of a different shape, then the neural network will be worse at determining the ownership of this object. Moreover, the more differences there are, the greater the chance of an incorrect definition of the object. This problem can be solved by increasing the variety of object shapes in a variety of training data.

6 Conclusions

Since the developed and trained convolutional neural network has been experimentally proven to successfully recognize 3D models, its further development makes it possible to obtain the result of processing not just one 3D model but also several 3D models located on the same image. By increasing the variety of possible forms and characteristics of training data, it is possible to improve the correct recognition of objects, even if they are very different from the usual representation of the training object.

Then, on the basis of the objects being determined, it is possible to determine various situations using imaginative thinking at the program level. As already mentioned, in order to move from the usual recognition of objects based on perception, in our case, visual, to recognition by finding various patterns, it is necessary to perform complex and extremely long work just to identify various patterns characteristic of a particular object. Neural networks are great for this kind of task. With properly structured neural network training, the time for this very training is enormously reduced if we compare the time for which a person and a computer will be able to determine a set of attributes related to a specific object.

In the future, it is planned to develop such an artificial intelligence system that will be able not only to recognize all the objects in the image but also to determine the situation that occurs at the exposition. For example, train the system to recognize not only the teapot but also the cups located next to it at the same time. After correct recognition, the program can make the assumption that in this situation, the contents of the teapot can be poured into mugs.

References

1. Shimonishi H., Murata M., Hasegawa G., Techasarntikul N.: Energy optimization of distributed video processing system using genetic algorithm with bayesian attractor model. In: Proceedings of 2023 IEEE 9th International Conference on Network Softwarization (NetSoft) (2023). https://doi.org/10.1109/NetSoft57336.2023.10175483
2. Cha, S.-H.: Comprehensive survey on distance/similarity measures between probability density functions. Int. J. Math. Models Methods Appl. Sci. **1**(4), 300–307 (2007)
3. Hamidreza Kasaei, S.: OrthographicNet: a deep transfer learning approach for 3D object recognition in open-ended domains. IEEE/ASME Trans. Mechatron. **26**(6), 2910–2921 (2021). https://doi.org/10.1109/TMECH.2020.3048433
4. Golovko, V., Kroshchanka, A., Mikhno, E.: Brands and caps labeling recognition in images using deep learning. In: Ablameyko, S.V., Krasnoproshin, V.V., Lukashevich, M.M. (eds.) PRIP 2019. CCIS, vol. 1055, pp. 35–51. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-35430-5_4
5. Anto Bennet, M., Srinath, R., Abirami, D., Thilagavathi, S., Soundarya, S., Yuvarani, R.: Performance evaluation of video surveillance using mete, melt and nidc technique. Int. J. Smart Sens. Intell. Syst. **10**(5), 25–54 (2022). https://doi.org/10.21307/ijssis-2017-234
6. Lukezic, A., Vojr, T., Zajc, L.C., Matas, J., Kristan, M.: Discriminative correlation filter tracker with channel and spatial reliability. Int. J. Comput. Vision **126**(7), 671–688 (2018)
7. Kochanov, A., Zolotukhin, V., Mironenko, V., Savelyeva, A., Polyakova, A.: Digital processing of satellite images using neural network algorithms. J. Phys. Conf. Ser. **2373**, 062026 (2022)