Building an artificial vision system of an agricultural robot based on the DarkNet system

M G Dorrer, A A Popov and A E Tolmacheva
Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, prospect "Krasnoyarsky rabochy" 31, Russia

E-mail: tolynbms@yandex.ru

Abstract. The article proposes a solution for the implementation of the artificial vision of an agricultural robot (farmbot). The Yolo3 system in the Darknet topology was chosen as the software platform. Recognition of objects (strawberries) and their discrimination with other objects in the farmbot workspace is provided by a convolutional network of deep learning with the primary use of the neuron activation function ReLU. The architecture used made it possible to ensure a percentage of correct recognition of objects up to 92 - 93% during dynamic processing of the video stream, including when processing objects that have a significant external similarity with the target object - strawberries. Similar results suggest that the farmbot computer vision system built on the Yolo3 platform in the Darknet topology will be fully operational.

1. Introduction
Following the automation of production, agriculture is also changing. Robots created for agricultural work are not inferior to industrial robots in their manufacturability. They can move around the cultivated area autonomously. To perform agricultural work, they use machine vision and artificial intelligence. Farmer robots can improve the efficiency of agriculture, attracting young specialists, new companies and new investors to the industry. The result is an increase in agricultural productivity, which is very important in the context of an increase in the world's population and an increase in food demand. Another important effect of agricultural robotics is the reduction of production costs. Work in the field of agricultural robotics is being actively carried out in the world, a special term is adopted for agricultural robots - “farmbot” [1-2].

For more than 40 years, artificial vision systems have been associated with the tools of artificial neural networks. Important results in this area were obtained back in 1975 by K. Fukushima, [3-4]. LeCun [5-6] made a significant contribution to this area, and active work is currently being carried out on the use of artificial neural networks in the artificial vision of robots [7-8].

In the field of computer vision, several software platforms have spread. TensorFlow [9] is a comprehensive open source machine learning platform. It was developed by the Google Brain team as a continuation of DistBelief's proprietary machine learning system.

PyTorch [10] is an open source Python machine learning environment that provides tensor computing with GPU acceleration. It was developed by Facebook and introduced in October 2016, and open to third-party developers in January 2017. The framework is suitable for rapid prototyping in research, as well as for lovers and small projects.
Keras [11] is an open deep learning environment written in Python. It was developed by Google engineer Francois Scholl and introduced in March 2015.

Darknet [12] is an open source framework written in C using the CUDA software and hardware parallel computing architecture. It is fast, easy and convenient to use. Darknet also supports calculations based on CPU and GPU. The framework is aimed at operational work with neural networks and is compact, modular and expandable.

Thus, the task of using artificial neural networks in the computer vision of agricultural robots - farmbots is relevant. To work in the conditions of a university laboratory with the aim of quickly obtaining a prototype of a pharmaceutical boat system for a pharmaceutical boat, the Darknet framework was chosen as a software platform.

2. Methods

Work with an artificial neural network includes the following steps that must be correctly passed for the system to work correctly.

The minimum image resolution suitable for training is 480p (640x480 pixels). The advantages of this resolution are the amount of information and, accordingly, the computational complexity. However, this will make recognition difficult, since this will reduce the number of characteristic features of an object for a neural network. If the hardware resources of the system allow you to process images with a resolution of 720p (1280x720), then it is better to work with this format, as the likelihood of correctly determining the object will increase. At the same time, training time and computing resources will increase simultaneously.

In the process of preparing data for training a neural network, it is necessary to use the markup program to assign an object class and indicate the coordinates of its sample. For high-quality training, it is necessary to select the area as close to the image as possible, otherwise the neural network will make mistakes more often, believing that the part of the background that got into the selection is also a fragment of the object.

To select a sample, it is necessary to consider each frame as a completely separate image. This helps to qualitatively mark the samples of the object, which improves the quality of the solution of the problem of determining objects by the neural network.

A ripe strawberry was chosen as an object identified by the computer vision system [9].

To ensure reliable training of an artificial neural network, a training dataset was prepared consisting of 4714 images with ripe strawberries and 2000 dandelion flower frames. The work used the architecture of the Yolo3 model in the Darknet topology.

The base topology is darknet-53, which was pre-trained for 2017 and includes 14197122 images, divided into 21841 classes.

The image arriving at the input of the convolutional network by default resizes (collapses) to 448x448 and passes through the modified GoogLeNet architecture [9]. The output is a property map (feature map), then several consecutive convolutions occur, during which there is a decrease in the dimension of the feature space. In the case of a network that distinguishes between 20 classes, the result of convolution is the tensor 7x7x30. This tensor provides a definition of classes. A 7x7 grid is superimposed on the original image. Each grid cell is associated with a 1030-dimensional vector. The first 10 numbers contain information about the coordinates and sizes of the two framing rectangles for the cell, as well as an indicator of the degree of confidence that the rectangles correctly found the detected object. The remaining 20 numbers correspond to the objects of classification and show confidence that the center of the object lies inside the cell, regardless of the various framing rectangles. Next, the confidence score for each cell is multiplied by the probability of each class to obtain the final score. The current version of the convolutional neural network Yolo v3 contains 75 convolutional layers.

A graphical representation of the model architecture is shown in figure 1.
Figure 1. General view of the architecture of the YOLOv3 model.

A fragment of the architecture of a computer vision neural network for an agricultural robot implemented on the basis of Yolo v3 is shown in figure 2.

Figure 2. An example of output nodes of the YOLOv3 architecture.
The implemented neural network uses the ReLU neuron activation function. This function has the form \( f(x) = \max(0, x) \), that is, for negative values it returns the result “0”, and in the positive half-plane it returns the input value to the output. Consider the positive and negative sides of this activation function.

Positive sides:

- Calculation of sigmoid and hyperbolic tangent requires resource-intensive operations such as exponentiation, while ReLU can be implemented using a simple threshold transformation of the activation matrix at zero.
- The ReLU function is not saturated.
- The use of ReLU significantly increases the convergence rate of stochastic gradient descent (in some cases up to 6 times compared with the sigmoid and hyperbolic tangent. It is believed that this is due to the linear nature and the lack of saturation of this function.

The negative side of the application of this function is that the neural networks built on the ReLU function are not always reliable enough and can fail (“die”) in the learning process. For example, at high values of the modulus of the gradient vector passing through ReLU, such a change in the weights of synapses can be obtained that this neuron is never activated again. If this happens, then, starting from this moment, the gradient passing through this neuron will always be zero. Accordingly, this neuron will be irreversibly incapacitated and excluded from further operation of the neural network. For example, if the learning rate is too high, it may turn out that up to 40% of neurons with the ReLU function are “dead”, that is, they are never activated. This problem is solved by choosing the appropriate learning rate.

Table 1 describes the technical characteristics of the computer on which the neural network was trained to determine strawberry berries.

| Type            | Name                                | Parameter                                      |
|-----------------|-------------------------------------|-----------------------------------------------|
| CPU             | Athlon 64 X2 3800+                  | Bus Frequency: 2000MHz Critical temperature: 70C CPU Frequency: 2.0 GHz |
| RAM             | KINGSTON HyperX FURY Black HX424C15FBK2/8 DDR4 | 16 Gb                                         |
| Video card      | nvidia gtx 1050 ti                  | GPU: GeForce GTX 1050 Ti, the amount of video memory: 4 Gb |
| HDD             | Seagate Barracuda ST1000DM010       | 1Tb                                           |
| Operating system| Linux Ubuntu                        | 16.4                                          |

Training a neural network to determine the ripe strawberries required 30,000 iterations. A neural network trained in this way correctly detected objects in 92% of cases.

3. Results
As a result of training, an artificial neural network detects strawberries in the training video with a minimum threshold of 50% probability and the maximum correct answer is determined with 99% probability.

The result of training a neural network is shown in figure 3.
Figure 3. The work of the neural network in the training video "Strawberry".

When an artificial neural network analyzes a random video, where there is nothing but strawberries, the percentage of the correct answer is also quite high and ranges from 61 to 93% (figure 4).

Figure 4. Neural network operation on random video.

For the control experiment, videos were taken with objects similar to strawberries. In the case of cherries, the color matches in bright light or unripe berries. In this case, a minimum detection threshold of 50% is set. Then the percentage of the correct answer is in the range of 58-92%.

You can also see the result of the operation of an artificial neural network in a video with similar objects. As an example, cherries were used as a similar object (figure 5).
Figure 5. An example of the operation of an artificial neural network with video strawberries with cherries.

If strawberries are highlighted in the image, where tulips are presented along with strawberries, situations of the same shape and color occur for the neural network if a red tulip lies next to the strawberries. To adequately identify the object, a minimum detection threshold of 70% was set. With these settings, the percentage of correct answers was 72-90%.

4. Discussion
We see that in order to reduce the likelihood of an error when isolating strawberries from cherry berries, the solution is to add color filters to better determine the object, select the minimum detection thresholds or, and also retrain the neural network to an additional class that corresponds to the most common interference objects.

To reduce the likelihood of a false positive in the neural network when separating red tulips from ripe strawberries, such solutions seemed effective - increasing the quality of the video image, adding detection filters, and also training the neural network to determine the additional class of “tulip”.

Thus, in order for an artificial neural network to be able to correctly identify objects and distinguish from incorrect ones, it must be trained to recognize not only target objects, but also objects similar to target ones and at the same time with a high probability of occurring in the working environment of a farmbot. In addition, the reliability of determining an object increases if the farmbot takes the time to examine an unknown object from different angles and then decide on assigning the object to any class. In addition, the use of graphics with the highest possible resolution increases the likelihood of the correct definition of objects.

5. Conclusion
Plants are a very complex subject for the analysis of artificial neural networks. Overcoming this problem, it is necessary to train the neural network for all the organs of each plant with which it will encounter during the work. This will take time and require enough processing power of the hardware, especially for the neural network training phase. However, only such an approach will achieve reliable recognition by the farmbot of objects in the served area.
References

[1] Bechara A and Vigneault C 2016 Agricultural robots for field operations: Concepts and components Biosystems Engineering 149 94-1

[2] Murdyantoro B, Sukma D, Atmaja E and Rachmat H 2019 Application design of farmbot based on Internet of Things (IoT) International Journal on Advanced Science, Engineering and Information Technology 9(4) 1163-70

[3] Fukushima K 1973 A model of associative memory in the brain, Kybernetik 12 58-63

[4] Fukushima K 1975 Cognitron: A self-organizing multilayered neural network Biological Cybernetics 20 121-36

[5] LeCun Y, Haffner P, Bottou L and Bengio Y 1999 Object Recognition with Gradient-Based Learning in Shape Contour and Grouping in Computer Vision. Lecture Notes in Computer Science (Springer, Berlin, Heidelberg) 1681 27

[6] LeCun Y, Bottou L, Bengio Y and Haffner P 1998 Gradient-based learning applied to document recognition Proceedings of the IEEE 86(11) 2278-324

[7] Frese U and Hirschmüller H 2015 Special issue on robot vision: what is robot vision? J Real-Time Image Proc 10 597-8

[8] Ruiz-del-Solar J and Loncomilla P 2020 Applications of Deep Learning in Robot Vision in Deep Learning in Computer Vision CRC Press 211-32

[9] An end-to-end open source machine learning platform 2019 Retrieved from: https://www.tensorflow.org/

[10] PyTorch 2020 Retrieved from: https://pytorch.org/

[11] Keras: The Python deep learning API Retrieved from: https://keras.io/

[12] Redmon J Darknet: Open Source Neural Networks in C Retrieved from: https://pjreddie.com/darknet/