# The use of machine learning algorithms for image recognition

Jan Matuszewski[*1], Adam Rajkowski[2]
[1]Military University of Technology, Faculty of Electronics, Institute of Radioelectronics, 2, gen. Sylwestra Kaliskiego St., 00–908 Warsaw, Poland; [2]Sandomierz Radoszki, JW 3533, 3 brt, Poland

## ABSTRACT

The article presents a way of using machine learning algorithms to recognize objects in images. To implement this task, an artificial neural network was used, which has a high adaptability and allows work with a very large set of input data. The neural network was described using a program written in the MATLAB simulation environment. The basic problem faced by the designer of objects recognition is to collect a sufficient training set of images to achieve the high probability of correct recognition. The set of learning patterns in the artificial neural networks may contain from several dozen thousands to one million training samples. In this article at the beginning the neural network was pre-trained trained based on the images included in the publicly available CIFAR 100 database, which are characterized by a small size of 32x32 pixels. It contains 70 000 images assigned to 10 basic categories. Then the author's database, consisting from 1000 pedestrians, cars and road signs was used. The article contains a description of applied algorithm, method of supervised learning and correction of weight coefficients, selection of activation function and operation on max pooling filter. The results of proposed solution are presented in the form of screenshots from calculations and in figures depicting results of recognized objects. Attention was also paid to the impact of used database for learning the network on the speed of calculations and recognition efficiency. The proper selection of number and types of layers, number of neurons, activation function and the value of the learning factor is very important in designing the neural network in application to objects recognition contained in the images. The problems occurring in the process of learning the neural networks and suggestions for their further improvement are also presented.

**Keywords:** image recognition, deep neural network, machine learning algorithms.

## 1. INTRODUCTION

The subject of the use of artificial intelligence, in situations where it is impossible to clearly classify data, is enjoying increasing popularity nowadays. Artificial neural networks (ANNN) are used in smartphones, autonomous cars or translational tools [1, 2, 11, 17]. Thanks to their ability to learn, based on searching for similarities between objects and their generalization, they are able to deal with problems where a very accurate classification is required [6, 8, 18, 19].

The article presents an application for recognizing objects in images using machine learning algorithms, which task is to recognize objects visible in the image and assigning them the correct label [13, 16 . Particularly important here is the high ability to identify objects that were not previously included in the training set, which distinguishes neural networks from other algorithms.

## 2. IMAGE ANALYSIS

Before it becomes possible to use the ANN as a model using machine learning algorithms, it is necessary to properly process the image, which can be divided into several stages as shown in Figure 1. Segmentation is the first activity carried out in the process of machine learning to recognize objects [3, 14].

The image is here divided into parts that are somehow related to each other. This is to pre-isolate areas that belong to a given objects, its boundaries, form or limit sending to the next stage of unnecessary information in the computer memory.

[1] jan.matuszewski@wat.edu.pl; phone +48 261837571; fax +48 261 837 461
[2] adamr_1995@wp.pl; phone +48 794577529

Radioelectronic Systems Conference 2019, edited by Piotr Kaniewski, Jan Matuszewski, Proc. of SPIE
Vol. 11442, 1144218 · © 2020 SPIE · CCC code: 0277-786X/20/$21 · doi: 10.1117/12.2565546

The next stage is the analysis of image features, which makes it possible to reveal and describe the object properties, which often remain unnoticed using only eyes.

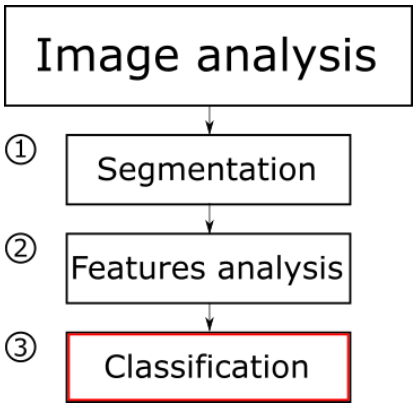


Figure 1. Image analysis diagram

Object features can be grouped into several basic groups, e.g. geometric, nongeometric, topological. The choice of which image features are to be analyzed is an individual matter, depending on the final result [4, 5, 10]. For the purposes of this algorithm, however, the most important is the mathematical side of the processing the analyzed image features, resulting in the so-called signatures and skeletons, that is, one-dimensional functions representing the contours of an object. After completing these preliminary actions, there is a process of object recognition using the artificial neural networks, in which it is possible to use the deep learning methods.

## 3. ARTIFICIAL NEURAL NETWORKS

### 3.1 Basic information

ANN is the name of mathematical structures and their software or hardware implementation, consisting of individual elements called neurons capable to performing basic operations at their input [17].

The principle of the neuron can be represented as follows:

$$e = \sum_{i=1}^{n} x_i w_i \tag{1}$$

$$y = f(e) \tag{2}$$

where: $x_i$ - value of the $i$-th input signal, $w_i$ - weight factor of the $i$-th signal, $n$ - number of neurons in the input signal, $e$ - total value of neuron stimulation, $y$ - value on the output neuron, $f$ - activation function.

After calculating the sum $e$, using the weight factors $w_i$ and input signals $x_i$, the result is multiplied by the activation function $f$, which should meet the following conditions:

- continuity of change between your minimum and maximum values,
- derivative continuity (the derivative should also be not difficult to calculate).

The ReLU activation function was used which parameters are depicted in Figure 2. This type of activation function is currently the most-used activation function for learning neural networks that are designed to recognize objects in images. This is due to her endlessly striving response for a valid signal and zeroing the neuron value for a negative signal.

Such approach means that not all neurons are used in the network, which protects it from *overfitting* and speeds up the process of network learning. In addition, *ReLU* calculates the derivative very simply, and its linearity enables the use of back error propagation method to make correction in weight coefficients in the network.

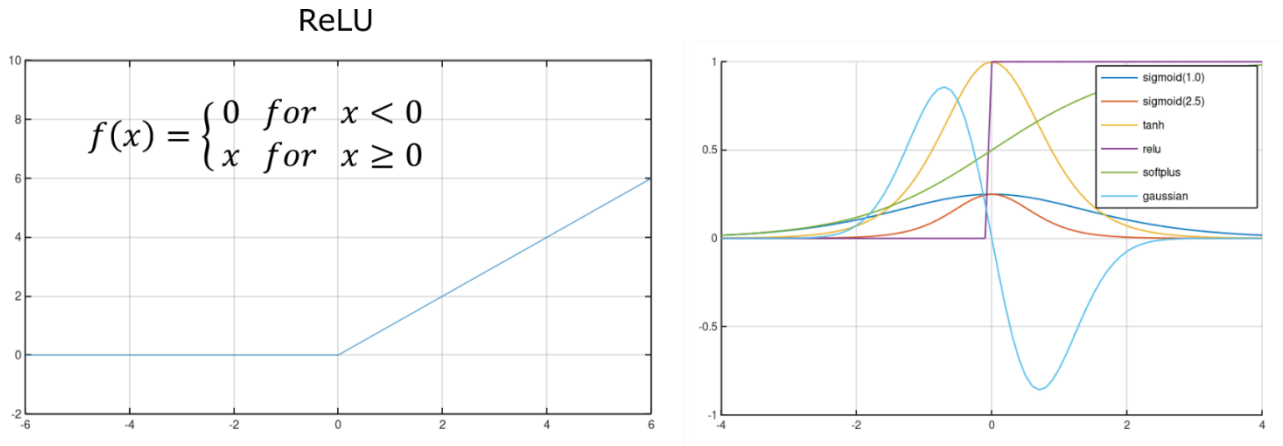


Figure 2. Presentation of the ReLu activation function and comparison of its first derivative (right) with derivatives of other activation functions

### 3.2 Neural network model

Creating a program for recognizing objects in the image requires the development of a mathematical model and a comprehensive approach due to the large amount of data necessary for processing and classification. One of the possible solutions for recognizing objects in the image, due to the effectiveness of operation, is the use of ANN, in which the correct machine learning algorithms will be implemented. The diagram of the neural network used in this program is depicted in Figure 3, [15].

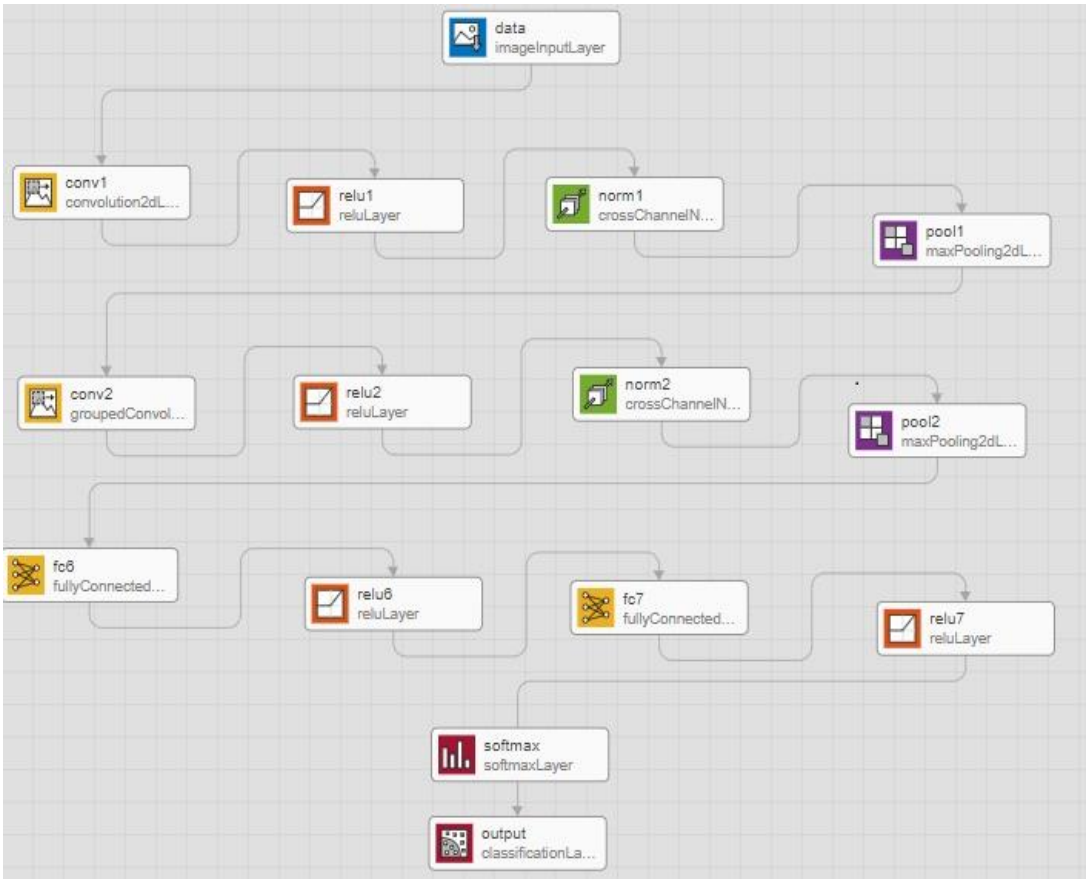


Figure 3. Proposed model of a neural network

This is a *feedforward* network. Information in this type of network architecture only moves forward, Fig.4. The network consists of several layers, which is a very common solution [20]. The result is a complex structure with a large number of connections, prone to so-called *overfitting*. The network consists of an input layer, hidden layers and an output layer, which by their nature can be further divided into two groups:

- The first three layers of the network (including the input layer), which, thanks to the use of filtering masks, adapted to the expected characteristics, reduce the "dimensionality" of the image, and then using the characteristic hierarchical pattern contained in the data (several characteristic features are responsible for shape recognition not the whole image because some pixels carry more information than others), allow the use of fewer neurons and connections between them to achieve the same effect. This is particularly important in the case of multi-layer networks susceptible to data overtraining.
- The layers with full connections of neurons to each other (this network is still one-way), which are the last three layers for data processing (including the output layer, with the number of neurons equal to the number of objects possible to classify) and two layers with 4096 neurons, which are responsible for mathematical calculations. Figure 4 shows the differences between the architecture of layers.

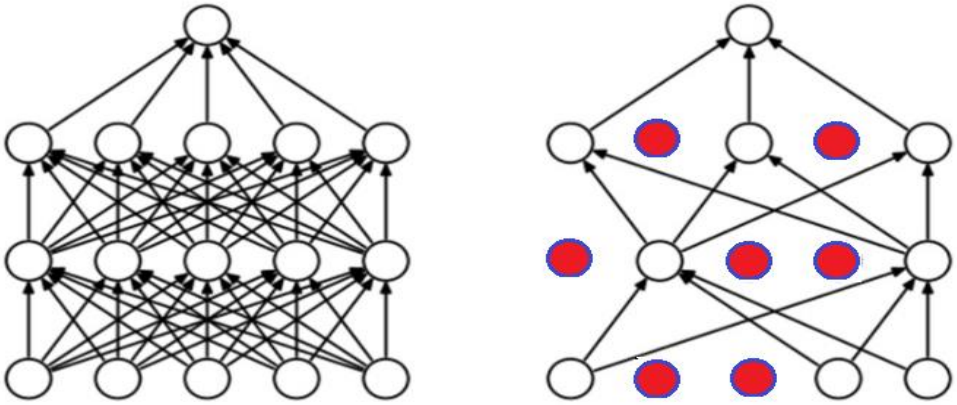

Figure 4. A network with fully connected layers (left) and a regularized network (right)

The use of this algorithm allows to accelerate the operation of the neural network, because the programmer acts as a teacher who knows what values he expects at the output of the neural network, determining the correctness of the algorithm result. When the discrepancy is too large, he gives a suggested change in value that will allow correct recognition of the object and instructs the algorithm to make the next iteration - the network learns knowing what result it should obtain, so initially random values on individual neurons will quickly set at the level enabling assumed network operation. The idea of the back propagation method is shown in Figure 5.

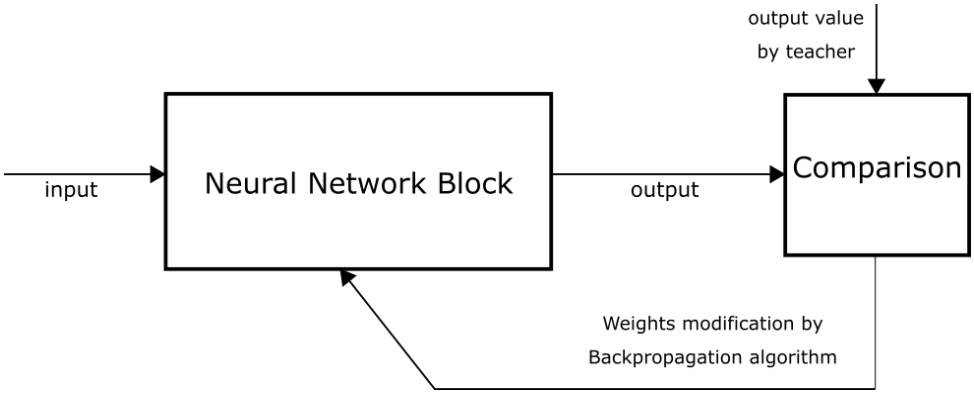


Figure 5. Idea of learning algorithm with teacher together with weight change by back propagation method

This method is used to calculation the neuron values - it does so using mathematical formulas. The basis of its operation is the use of knowledge about the result to be obtained at the output of the network. Then the error is calculated between the suggested value and that obtained by the network and the error is corrected by changing the value on neurons [11]. This simple task is complicated by a network consisting of several layers, which is the basic type used in more advanced artificial intelligence algorithms. In this case, the back propagation algorithm sums up the neuron errors from the hidden layer preceding the last modified layer and only then corrects their values and weights between them.

The operations of the algorithm can be represented as follows:

- Randomizing weight values and assigning them to a neuron.
- Loading the input the data (already in mathematical form).
- Input of the suggested output value and on its basis the calculation of the values at the outputs of neurons and their comparison. Calculation of network output error.
- Calculation of errors in the hidden layer behind the output layer, taking into account the error of the output layer and the sum of the hidden layer error (maintaining the weights between neurons).
- Repeating the procedure for the next hidden layer, using its summed errors and the output error preceding the hidden value calculated in the previous step.
- Repeating the procedure for the next hidden layer, using its total errors and the output error from the preceding hidden layer, calculated in the previous step.
- After calculating the errors in all hidden layers, the algorithm changes the values of neurons on the input layer in the same way.
- All weights in the network are modified.
- The error decreases with each complete pass of the algorithm until it falls to the teacher's acceptable level.

## 4. OPTIMIZATION OF NETWORK MODEL OPERATION

The neural network model presented in section 3.2 consists of several basic elements. In addition to using its typical neural network layers consisting of neurons and activation functions, it also has elements responsible for the optimization of the network's operation, which allows its proper calculations and blocks the possibility of overtraining, despite a very large amount of data.

The model's function *norm1* is responsible for the *local response normalization*, through the procedure so-called attenuation, whose task is to normalize the infinite activity of the neuron. Two types of normalization are possible, each of which tends to strengthen the excited neuron and suppress neighboring in the range of limited values. The method searches for the strongest neuron responses and normalizes the responses of neighboring neurons, making the selected neuron even more sensitive to object features. If all the responses of neighboring neurons are large enough, then the function will limit the values in all neurons from a given channel, because they will not accept any of them to be particularly sensitive. In addition to normalization, this function also limits the number of neurons used in the learning.

There are two types of normalization: in the same channel (a group of neighboring neurons) and between the channels, which involves considering the neighborhoods in three, but not in two dimensions.

Below is the formula describing the process of two-dimensional normalization, i.e. for neighboring groups of neurons:

$$b_{x,y}^{i} = a_{x,y}^{i} / (k + \alpha \sum_{j=\max\left(0,i-\frac{n}{2}\right)}^{j=\min\left(N-1,i+\frac{n}{2}\right)} {a_{x,y}^{j}}^{2})^{3} \quad (3)$$

where: $b_{x,y}^{i}$ - normalized output of the kernel "*i*" (weight summation place) at position (*x*, *y*), $a_{x,y}^{i}$ - kernel source output attached to position (*x*, *y*), *N* - total number of neurons, *n* - normalization channel size, *k* - function hyperparameter.

After normalization, the data is being sent to the statistical filter so-called *max pooling*, which extracts the maximum value from the mask and reduces the number of calculations in subsequent layers. The application of this filter to non-overlapping subregions results in passing only the largest values, which significantly reduces the amount of data necessary for further processing, without reducing the effectiveness of the algorithm [9]. The simplicity of this solution for the 2x2 masking filter is shown in Figure 6.

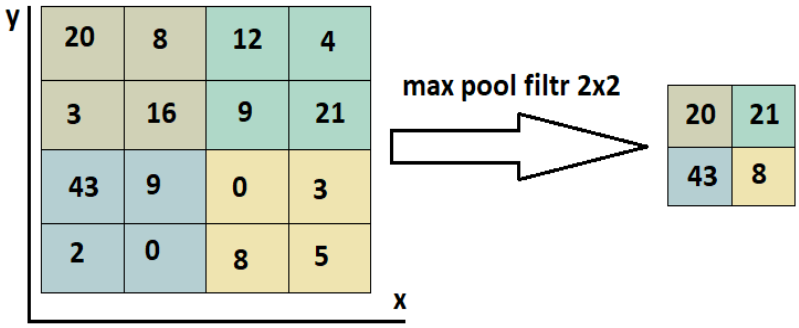


Figure 6. Operation of max pooling filter

Before the information reaches the network output, it still has to go through the *Softmax* function. Its task, as a transfer function, is to change the input vector from the received value into the output information in the form of a vector with normalized values between 0 and 1, so that the output layer receives specific information that can be interpreted as a certain probability, which will help determine the percentage of accuracy networks in case of its analysis. An example of how this function works is shown in Figure 7.

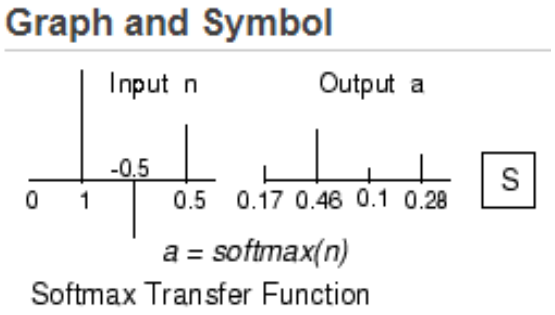


Figure 7. Operation of Softmax function

In adjusting the network structure help a such tools as *Optimal Brain Damage* and *Optimal Brain Surgery*, which tasks are detection and getting rid of neurons not involved in the operation. The most advanced Google algorithms for object recognition use the patented *Dropout* method, working similarly to the above-mentioned tools.

Cleaning the network not only ensures faster learning process, but also increases its accuracy, once again making it more resistant to *overfitting*. You should also remember to keep the optimal number of hidden layers, because too many of them will overwhelm the network, and too small may not be complex enough to solve the task.

## 5. THE PROCESS OF LEARNING A NEURON NETWORK

It is extremely important to set the *learning rate* at the appropriate level in the program [15, 17]. This factor defines by what maximum value can change the neuron weight. When this value is too low, the network learning process is disproportionately longer because it requires much more iterations to make the teacher-specified corrections. However, when it is too high, the weight correction performed on neurons will be so large that it will prevent the correct learning process.

The weights correction with large values means that the calculated values for neurons, despite subsequent iterations, will not be able to fit into the network scheme, which will cause them to be random all the time, such as in the initial phase of the learning process. When choosing the right *learning rate* (Fig. 8), the network learns correctly and after a few epochs already achieves the required recognition accuracy. Further learning has no greater impact on the improving recognition accuracy.

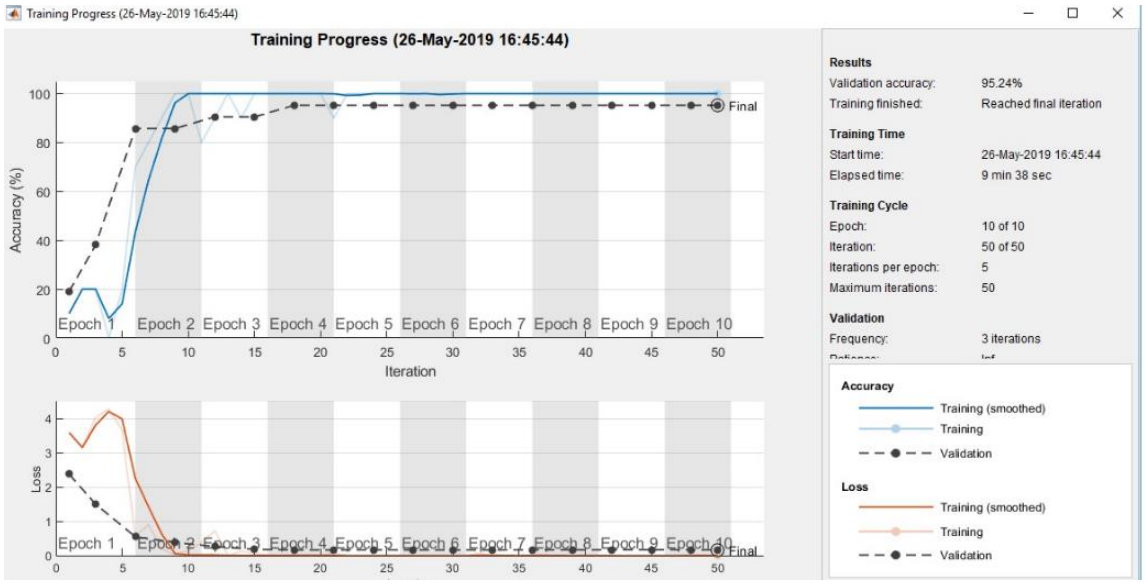


Figure 8. Network test for learning rate =1

If the value of this factor is set too high (Fig. 9), the network learning process is very fast, which is not always a positive phenomenon, because the network can stop at the local minimum and thus will not achieve maximum recognition accuracy. Incorrect selection of weights in the process of learning a network can also extend its learning time.

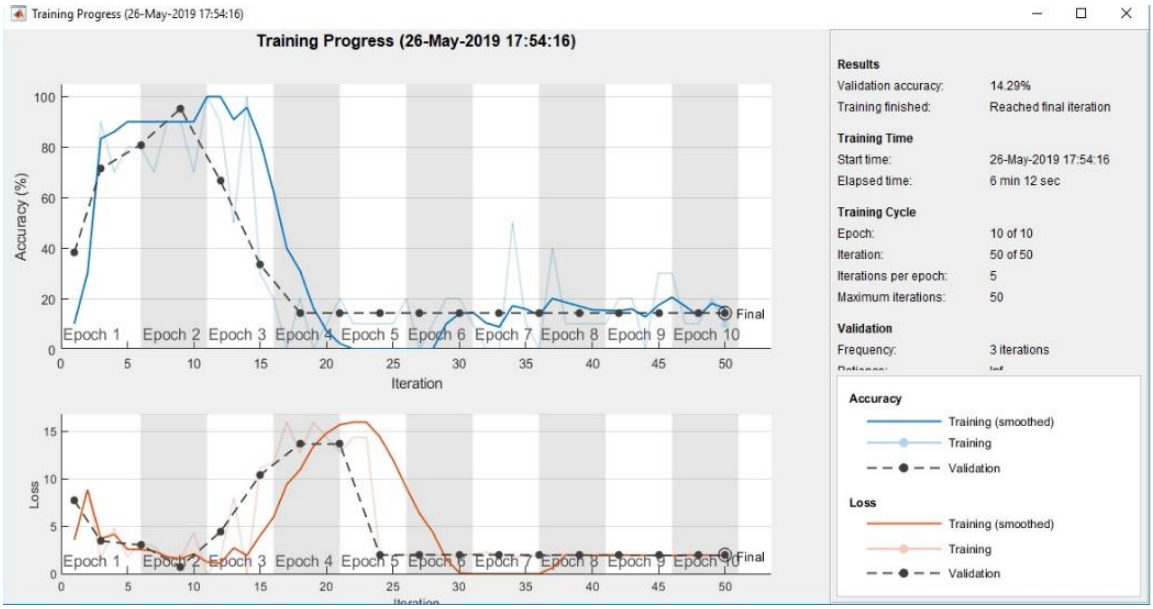


Figure 9. Network test for learning rate = 1

In the final phase, when the ratio has decreased due to the use of the *Drop LearnigFactor* function, the network is stuck on minimum accuracy values and will not work properly. Too little correction of weights, as shown in Figure 10, will unnecessarily extend the duration of the learning process.

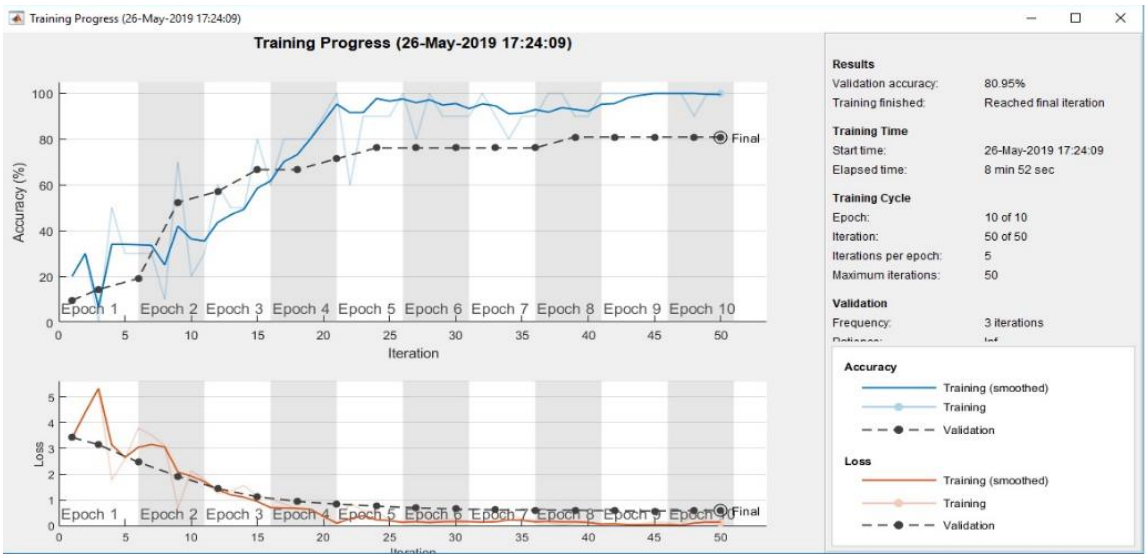


Figure 10. Network test for learning rate = 1

As you can see from the above calculation results, the worst for the network is setting too high weight correction, which can cause the mean square error to accidentally set below the value, despite the fact that the weight distribution on the neurons will not yet guarantee the correct operation of the network. To prevent such a situation, the momentum rule was used in the developed software application. It works by coupling together the correction values of weights from the previous and current iteration, through the factor $\alpha$. Such use the momentum causes that the weights correction (*learning rate*) starts from a certain high level, and then decreases with each iteration as it approaches the value specified by the teacher at the output. This approach extends the learning process and refines it and at the same time protects against accidental ending of the program due to a lot number of weights correction.

The method of using the parameter $\alpha$ is represented by the formula (4), which describes the change in weights used in the gradient method

$$\Delta w_i^t = -\eta \frac{\partial Q^t}{\partial w_i} + \alpha \Delta w_i^{t-1} \tag{4}$$

where: $\Delta w_i^t$ - modification of weight *i* after iteration *t*, $\eta$ - coefficient of change in the value of neuron weights (*learning rate*), $\frac{\partial Q^t}{\partial w_i}$ – gradient fall, $\alpha$ - weight correction factor from previous and current iteration.

The process of network learning depends on the correct use of all the above algorithms and their methods. For the network to work properly, it needs to provide a lot of training data, while using as few neurons as possible and connections between them to avoid overtraining.

## 6. RESULTS OF NUMERICAL CALCULATIONS

Training a neural network requires operation on a large set of input data. Creating such a large image database is extremely time consuming, therefore in the first stage of work the CIFAR-10 image database available on the Internet will be used (Fig. 11), [21]. Research and simulation with using the designed program was carried out in MATLAB environment, [12]. It contains 70 000 images of sizes 32x32 pixels, assigned to 10 basic categories, on the basis of which the network will be pre-trained. In the first stage, after loading the data into the network, they were scaled up, taking place in the input data layer, which consists of two basic parts:

- Image scaling. The final neural network will be adapted to recognize images with a strictly defined number of pixels 256x256, all larger images will be scaled up by cutting out nonexistent pixel features, while smaller images will be supplemented with Gaussian noise consisting of random white and black image placement pixels, which even by increasing the size of the file by a factor of 10, do not have a negative impact on the recognition of the image's characteristics.

- Creating the mirror images. The neural network is not able to distinguish the same object in different positions relative to the plane, because they are different objects for it, so during learning each object must be presented with networks from different perspectives and in different settings. This necessity arises from the fact that the network first looks for the simplest features in the object which it identifies. If is learned a group of objects facing to the right, e.g. chairs, and then a similar group of objects, but facing only to the left, then in the process of testing the network, giving her chair for classification, facing to left, the network would answer that it is a nightstand - because the most characteristic feature of nightstands would be their return to it. Therefore, it is necessary to "rotate" the subject in the learning process.

After uploading and training images to the network, it is able to recognize objects in images with high accuracy, giving them the right label. The network compares its recognition result with the real image and when the recognition result is correct, displays the label above that image in green and otherwise in red color. An example of how the above application works are shown in Figure 11, [15].

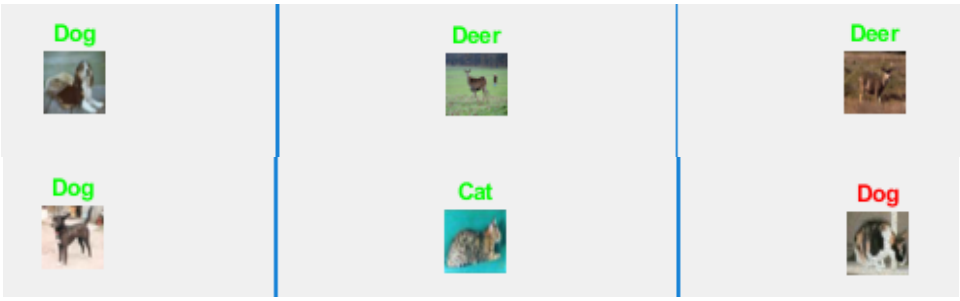


Figure 11. The result of recognizing test images using a neural network

Adding the ability of scaling the data enabled the use of the same database for a neural network with a slightly different structure, adapted to recognize images after adding the Gaussian noise in order to increase the number of images. Thanks to the initial testing of the network on images with lower resolution, the network worked faster and it took much less time to determine its structure and select the appropriate learning factor. This time, the percentage of the recognition correctness determined by the network is displayed above the images, Fig. 12.


Figure 12. Result of network operation

After testing the network operation in simpler cases, its subsequent application was presented, this time in connection with objects detection. The proven network structure (except the change of neurons number in the output layer to three - due to three categories of objects) remained unchanged, while the database consisting of 1000 pedestrians, cars and road signs

has been changed. The MATLAB function *rcnnObjectDetector* is responsible for the objects detection in the images, and the artificial neural network from previous points was loaded into it and trained on a new database [15].

Figures 13 and 14 show the result of calculations obtained from the program described above for pedestrian, car and road signs recognition in the image. After ending the learning process, the network correctly detected and recognized cars, pedestrians and traffic signs, Fig. 14.

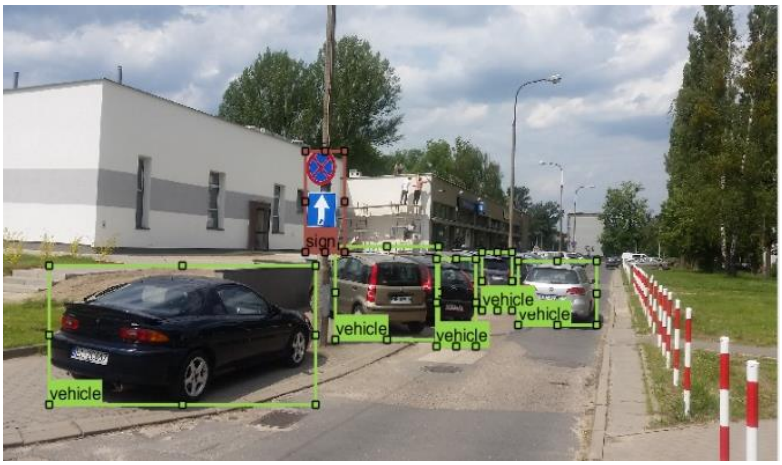


Figure 13. The result of vehicles and road sign recognition with using the neural network


Figure 14. The result of pedestrians, vehicles and road signs recognition with using the neural network

## 7. CONCLUSION

To sum up, the presented above application enables the different objects recognition in images, applying the machine learning algorithms for classification with using the artificial neural networks. The neural network is an excellent tool for recognizing objects in images, but it should remember about the appropriate selection of its model. The proper selection of number and types of layers, number of neurons, activation functions and the value of the learning factor is also extremely important. The interface of the computer application is based on the use of the *Deep Learning Toolbox tool*, enabling easy uploading to the previously designed neural networks, databases, or selecting the percentage of data to be used in the process of network learning, object recognition and validation.

The presented experiment results show the advantages of the software used in ANN processing. Knowledge of the neural network architecture allows to decrease the learning time and recognize objects in images in the final system in real time. In order to be able to use the ANN for recognition in real systems, it is necessary to have a large number of known reference objects, which can be used in the images learning process. The recognition model presented here can be adapted to the needs of any objects recognition.

## REFERENCES


[1] Chen, S.;Wang, H.; Xu, F. and Jin, Y.Q., "Target Classification Using the Deep Convolutional Networks for SAR Images," IEEE Transactions on Geoscience and Remote Sensing, 54, 4806–4817 (2016).

[2] Ciresan D. C., Meier U., and Schmidhuber J., "Multi-column Deep Neural Networks for Image Classification," IEEE Conf. on Computer Vision and Pattern Recognition CVPR (2012).

[3] Hryvachevskyi, A., Prudyus, I., Lazko, L. and Fabirovskyy, S., "Improvement of segmentation quality of multispectral images by increasing resolution," 2nd International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2017 - Proceedings 8095371, DOI: 10.1109/UkrMiCo.2017.8095371 (2017).

[4] Kaniewski P., Leśnik C., Serafin P.and Łabowski M., "Chosen Results of Flight Tests of WATSAR System," 17th International Radar Symposium IRS 2016, Krakow, 1-5, (10-12 May 2016).

[5] Kaniewski P., Kazubek J. and Kraszewski T., "Application of UWB Modules in Indoor Navigation System," 2017 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS), Tel Aviv, Israel (13-15 November 2017.

[6] Kirichenko L., Radivilova T.and Bulakh V., „Binary Classification of Fractal Time Series by Machine Learning Methods," In: Lytvynenko V., Babichev S., Wójcik W., Vynokurova O., Vyshemyrskaya S., (eds), Lecture Notes in Computational Intelligence and Decision Making. ISDMCI 2019. Advances in Intelligent Systems and Computing, vol 1020, Springer, Cham, 701-711, doi.org/10.1007/978-3-030-26474-1_49 (2020).

[7] Konatowski, S., "The development of nonlinear filtering algorithms," Przeglad Elektrotechniczny, Vol. 86, Issue 9, 272-277 (2010).

[8] Kraszewski T. and Czopik G., "Tracking of land vehicle motion with the use of distance measurements," Proc. of SPIE, 11055, XII Conference on Reconnaissance and Electronic Warfare Systems, 110550Y; DOI: 10.1117/12.2524954 (27 March 2019).

[9] Kraszewski T. and Czopik G., "Nonlinear Kalman filtering in the presence of additive noise," Proc. of SPIE, 10418, XI Conference on Reconnaissance and Electronic Warfare Systems, 104180N; DOI: 10.1117/12.2269355 (20 April 2017).

[10] Łabowski M., Kaniewski P. and Konatowski S., "Estimation of Flight Path Deviations for SAR Radar Installed on UAV," Metrology and Measurement Systems, Vol. 23, No. 3, 383–391, DOI: 10.1515/mms-2016-0034 (2016).

[11] Osowski S., [Sieci neuronowe do przetwarzania informacji], Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa (2006).

[12] Parrallel Neural Network Training with OpenCL: https://bib.irb.hr/datoteka/584308.MIPRO_2011_Nenad.pdf (27 November 2018).

[13] Pietkiewicz T., "A method of recognition of maritime objects based on FLIR (forward looking infra-red) sensor images using dynamic time warping," Proc. of SPIE 10434, Electro-Optical Remote Sensing XI; 1043409, doi.org/10.1117/12.2278419 (2017).

[14] Prudyus, I. and Hryvachevskyi, A., "Image segmentation based on cluster analysis of multispectral monitoring data," Modern Problems of Radio Engineering, Telecommunications and Computer Science, Proc. of the 13th International Conference on TCSET 2016, 7452020, 226-229, DOI: 10.1109/TCSET.2016.7452020 (2016).

[15] Rajkowski, A., [Wykorzystanie algorytmów uczenia maszynowego do rozpoznawania obiektów na obrazach], Praca dyplomowa, WAT, Warszawa 2019.

[16] Rogers, S.K., Colombi, J.M., Martin, C.E. and Gainey, J.C., "Neural networks for automatic target recognition," Neural Networks, 8, 1153–1184, (1995).

[17] Tadeusiewicz R. and Szaleniec M., [Leksykon Sieci Neuronowych]. Wydawnictwo Fundacji Projekt Nauka, Wrocław (2015).

[18] Vitalii, B., Kirichenko, L. and Radivilova, T., "Classification of multifractal time series by decision tree methods," 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. CEUR Workshop Proceedings 2015, Kiev; Ukraine, 457-460 (2015).

[19] Wajszczyk B. and Biernacki K., "Optimization of the efficiency of search operations in the relational database of radio electronic systems," Proc. of SPIE - The International Society for Optical Engineering, 2017 Radioelectronic Systems Conference, 107150H, DOI: 10.1117/12.2317741 0715 (19 April 2018).

[20] http://docplayer.pl/13483611-Sieci-neuronowe-wprowadzenie-agnieszka-nowak-brzezinska.html (05.05.2019).

[21] https://www.cs.toronto.edu/~kriz/cifar.html (access 12.05.2019).