UNIWERSYTET TECHNOLOGICZNO-PRZYRODNICZY
IM. JANA I JĘDRZEJA ŚNIADECKICH W BYDGOSZCZY
ZESZYTY NAUKOWE NR 261
TELEKOMUNIKACJA I ELEKTRONIKA 16 (2012), 61-74

# OBJECTS RECOGNITION SYSTEM WITH NEURAL NETWORK

Marcin Szczegielniak, Damian Szczegielniak

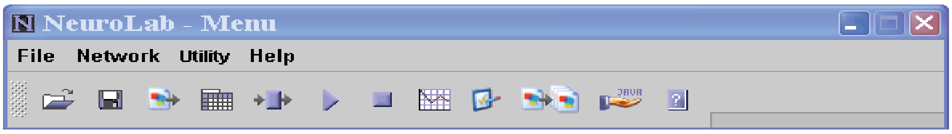
University of Technology and Life Sciences,
Kaliskiego 7, Bydgoszcz 85-796 Poland
szczegielniakm@wp.pl, dszczeg@utp.edu.pl

*Summary*: The system to recognize objects was designed and tested in the specialized application created and implemented by authors for neural networks and image processing simultaneously. This paper covers mathematical foundations of applied images standardization to achieve versatile and the invariant system to objects transformations (translation, scaling, rotation) in input images. Experiments with photos of objects were carried out and results were presented and discussed below.

Keywords: object recognition, image processing, neural network

## 1. INTRODUCTION

Artificial neural networks are used in many aspects or stages of the objects recognition. They can be applied not only as a classifier but for initial processing, segmentation or extracting objects features [7, 8]. There are many different types of artificial neural networks from classical structures to the newest ones which are still being developed [1, 6, 7]. However, this paper focuses on the versatile and popular multilayer feedforward perceptron as the features extractor and classifier simultaneously.



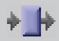
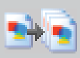

| | | |
|---|---|---|
| **Network** | | |
|  | Design Network | Window for designing the structure of the neural network. |
|  | Define Image Processing | Windows for designing an image processing. |
|  | Training Set | Window which allows to define a training set. |
|  | Training Set Average Error | Graph of the mean square error for the training set. |
|  | Test Trained Network | Window to test trained and ready networks. |
| **Utility** | | |
|  | Source Code Generator | It generates the Java code for a trained network (including designed image processing). This code can be used in any another Java application. |
|  | Image Processing Toolbox | It opens the window of the image processing toolbox. |

Fig. 1. Some functions available in the created application

The created specialized aplication allows for a robust and effective training of the network as well as to define a necessary initial images processing for the object recognition system. Although the aim of this paper is not to describe authors' application, Figures 1 and 2 may give only an idea how the environment of carried out experiments looks like. Further information is presented in next sections.

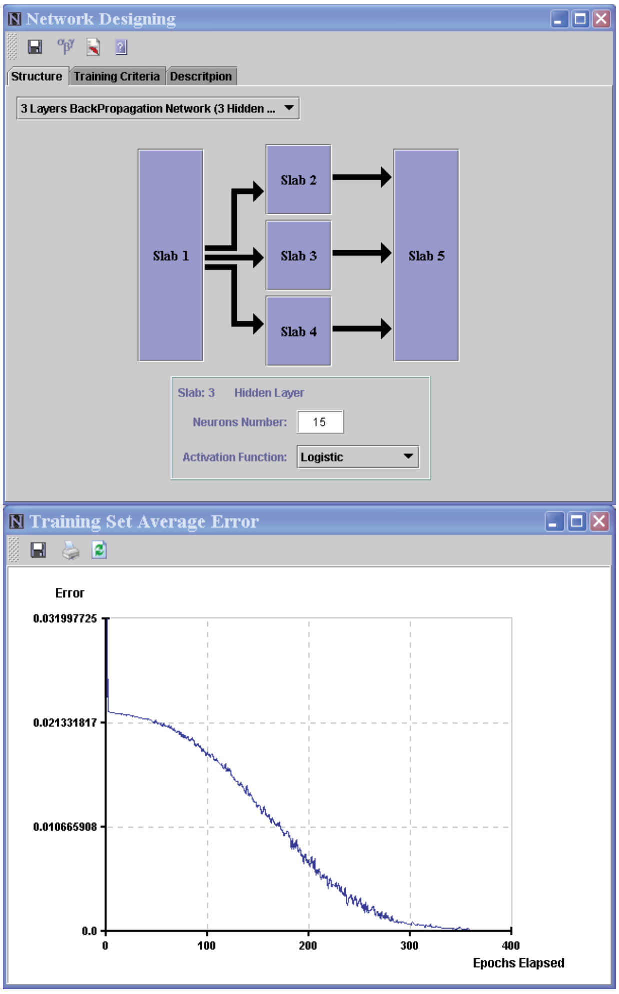



Fig. 2. Network designing and error preview during a training process

## 2. STANDARDIZATION OF INPUT IMAGES

It was assumed that a neural network was to extract some features of recognized input images on its own. Therefore, it is necessary to apply a proper initial processing which would allow a neural network to learn and work efficiently. It seems to be obvious that all possible variations of an object to recognize (with different sizes, rotations and positions in the input image) are too big set and should be reduced in a way before passing this image to the neural network. Therefore, a standardization of input images which makes an object independent of its translation (in X and Y directions), rotation and scale in the input image scene is executed first.

One of possible solutions [6] relies on cascade processing depicted in Figure 3. It is worth noting that the order of particular operations described below is important. This order allows to avoid exceeding bigger objects in the image beyond image boundaries during the rotation.

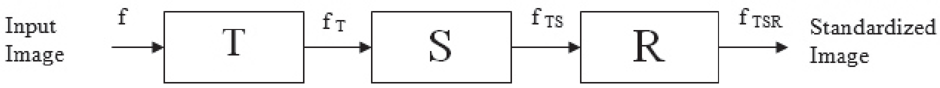



Fig. 3. Standardization of the input image which makes an object in processed images invariant to the translation (T block), scaling (S block) and rotation (R block)

The standardization T (the translation block) relies on finding a center of gravity of an object in the image and then moving it to the origin (assumed as a central point of an image). Thanks to this we can make the processed image translation-invariant [3].

We can calculate the center of gravity of a binary image in the following way

$$x_m = \frac{1}{P}\sum_{i=1}^{N}\sum_{j=1}^{N} x_i \cdot f(x_i, y_j) \tag{1}$$

$$y_m = \frac{1}{P}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i \cdot f(x_i, y_j) \tag{2}$$

where

$$P = \sum_{i=1}^{N}\sum_{j=1}^{N} f(x_i, y_j) \tag{3}$$

is the number of pixels with the value 1. The function $f(x_i,y_j)$ is the binary image with values 0 for the background and 1 for the object. We can describe this operation as

$$f_T(x_i, y_j) = f_T(x_i + x_m, y_j + y_m) \tag{4}$$

The aim of the scaling (S) block is to make the input image scale-invariant, a recognized object should be always the same size after this processing. It changes the size of an object in order to make the average distance between the origin and 1-valued pixels equal of a fraction of an average size of the image frame. The average distance can be described by

$$r_m = \frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{N} f_T(x_i, y_j)} \sum_{i=1}^{N}\sum_{j=1}^{N} f_T(x_i, y_j)\sqrt{x_i^2 + y_j^2} \tag{5}$$

A scale factor is specified by following equation

$$S = \frac{r_m}{R} \tag{6}$$

where R is an assumed fraction of the size of the image frame. This factor has to be chosen in order to avoid exceeding objects in the image beyond image boundaries (taking into account the next step - rotation as well). We assumed the value ¼ of this parameter. This operation can be described as

$$f_{TS}(x_i, y_j) = f_T(S \cdot x_i, S \cdot y_j) \tag{7}$$

The aim of the last rotation (R) block is to rotate an object to a standard/canonical position and make the processed image rotation-invariant simultaneously. To be more precise, this processing rotates an object in order to cover the direction of the maximum covariance with x-axis. For the two-dimensional image the covariance matrix is

$$C = \left( \frac{1}{P} \sum_{i=1}^{N} \sum_{j=1}^{N} f_{TS}(x_i, y_j) \begin{bmatrix} x_i \\ y_j \end{bmatrix} \begin{bmatrix} x_i \\ y_j \end{bmatrix}^T \right) - \begin{bmatrix} m_x \\ m_y \end{bmatrix} \begin{bmatrix} m_x \\ m_y \end{bmatrix}^T \tag{8}$$

where

$$P = \sum_{i=1}^{N} \sum_{j=1}^{N} f_{TS}(x_i, y_j) \tag{9}$$

and

$$\begin{bmatrix} m_x \\ m_y \end{bmatrix} = \frac{1}{P} \sum_{i=1}^{N} \sum_{j=1}^{N} f_{TS}(x_i, y_j) \begin{bmatrix} x_i \\ y_j \end{bmatrix} \tag{10}$$

Because $m_x$ and $m_y$ are equal zero (thanks to T block, the result of moving the center of gravity to the origin), the covariance matrix after neglecting the factor P is

$$C = \begin{bmatrix} T_{xx} & T_{xy} \\ T_{xy} & T_{yy} \end{bmatrix} \tag{11}$$

where

$$T_{xx} = \sum_{i=1}^{N} \sum_{j=1}^{N} x_i^2 \cdot f_{TS}(x_i, y_j) \tag{12}$$

$$T_{yy} = \sum_{i=1}^{N} \sum_{j=1}^{N} y_j^2 \cdot f_{TS}(x_i, y_j) \tag{13}$$

$$T_{xy} = \sum_{i=1}^{N} \sum_{j=1}^{N} x_i \cdot y_j \cdot f_{TS}(x_i, y_j) \tag{14}$$

The matrix eigenvalues C can be described by

$$\lambda_{1/2} = \frac{1}{2}\left[T_{yy} + T_{xx} \pm \sqrt{(T_{yy} + T_{xx})^2 - 4(+T_{yy} \cdot T_{xx} + T_{xy}^2)}\right] \tag{15}$$

The eigenvector connected with the matrix eigenvalue is

$$C\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \lambda \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{16}$$

On the basis of the equation 15 we get the gradient of the eigenvector

$$\frac{v_1}{v_2} = \frac{\lambda - T_{xx}}{T_{xy}} \tag{17}$$

Taking into account equestions 15 and 17 we have then

$$\frac{v_1}{v_2} = \frac{T_{yy} - T_{xx} + \sqrt{(T_{yy} - T_{xx})^2 - 4T_{xy}^2}}{2T_{xy}} \tag{18}$$

Functions $\sin(\Theta)$ and $\cos(\Theta)$, corresponding this gradient, are respectively

$$\sin(\Theta) = \frac{v_2}{\sqrt{v_1^2 + v_2^2}} = \frac{T_{yy} - T_{xx} + \sqrt{(T_{yy} - T_{xx})^2 - 4T_{xy}^2}}{M} \tag{19}$$

$$\cos(\Theta) = \frac{v_1}{\sqrt{v_1^2 + v_2^2}} = \frac{2T_{xy}}{M} \tag{20}$$

where

$$M = \sqrt{8T_{xy}^2 + 2(T_{yy} - T_{xx})^2 + 2(T_{yy} - T_{xx})\sqrt{(T_{yy} - T_{xx})^2 - 4T_{xy}^2}} \tag{21}$$

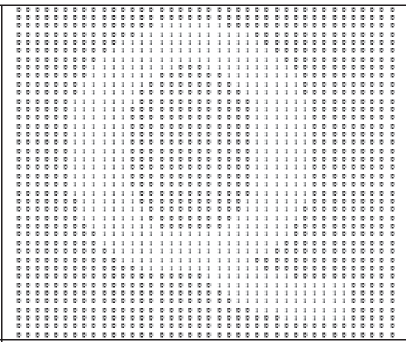
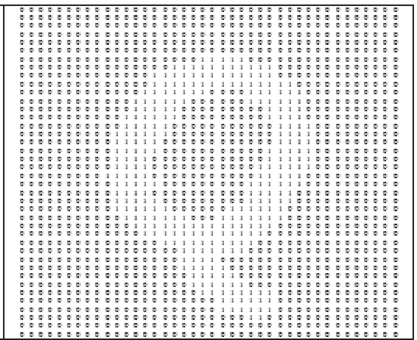
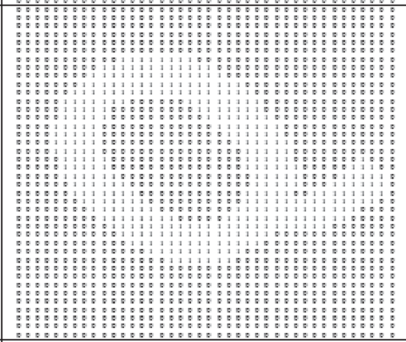
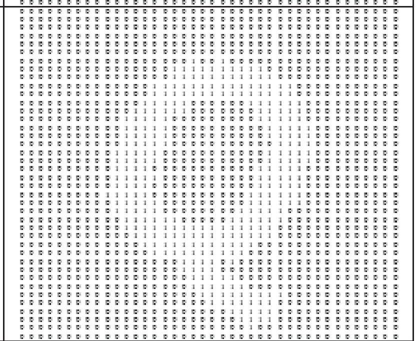
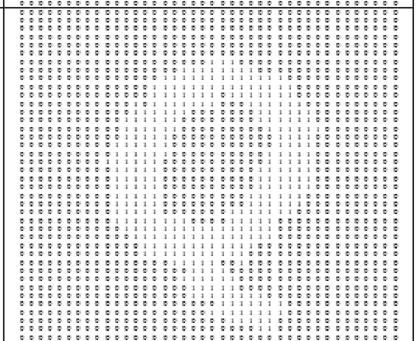
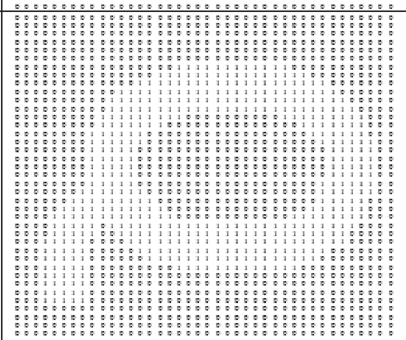
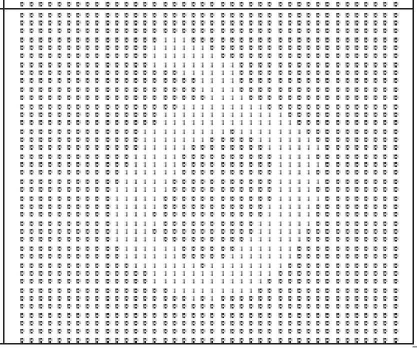
To sum up, the function projecting the image $f_{TS}$ into $f_{TSR}$ becomes

$$f_{TSR}(x_i, y_j) = f_{TS}(\cos(\Theta) \cdot x_i - \sin(\Theta) \cdot y_j, \sin(\Theta) \cdot x_i - \cos(\Theta) \cdot y_j) \tag{22}$$

It is worth mentioning that this projection doesn't recognize the direction and therefore, depending on the original object position in the image, the processed image $f_{TSR}$ can take one of two possible canonical positions, representing the same object. It has to be taken into account during the preparation of training patterns for the neural network.

In the table 1 there were a few examples of the standardization of an image described above. The standardization in the last case Q+90°.jpg resulted in the second possible canonical object position.

Table 1. Examples of the image standardization (input images in the left column and resultant standardized images in the right one)

| | | |
|---|---|---|
| Q.jpg |  |  |
| Q-45 °.jpg |  |  |
| Qsmall.jpg |  |  |
| Q+90 °.jpg |  |  |

## 3. RECOGNITION SYSTEM WITH NEURAL NETWORK

The proposed recognition system with the neural network was depicted in Fig. 4. A binary input image is standardized according to the processing described in section 3. Thanks to this an object in the standardized image can only be in one of two possible canonical positions which are invariant to the translation, scale and rotation of an object in the input image. Then the standardized image with the size MxM is reduced to NxN matrix, where N<M. The aim of scaling down the image is to reduce the structure of the neural network and further computational complexity. The input signal of the neural network is created by passing the input NxN matrix in the proper way (Fig. 4). It yields the input vector with $N^2$ elements.

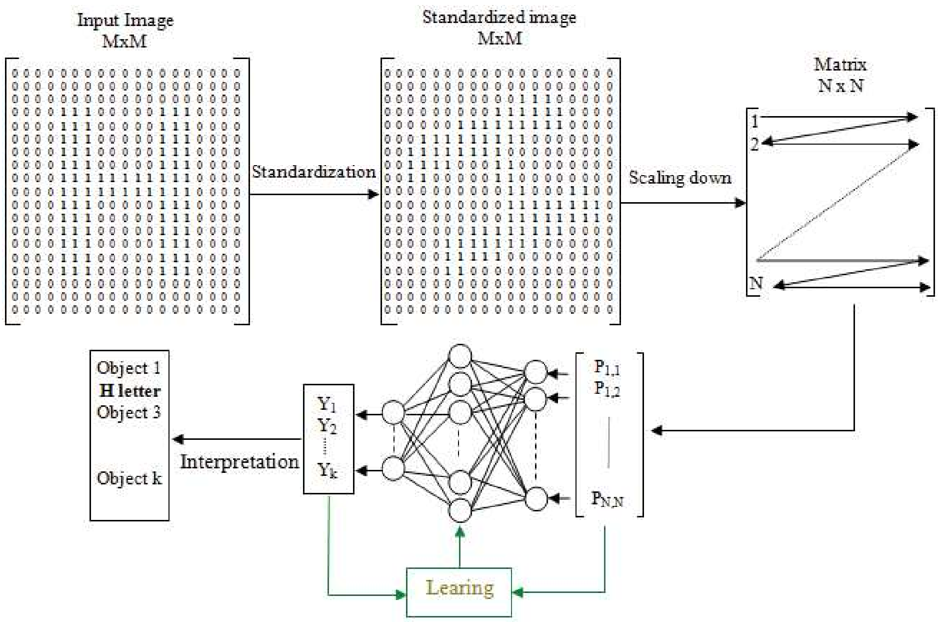



Fig. 4. Recognition system scheme

The number of neural network outputs was constant and depends on the number of recognized classes. The number of hidden neurons was only chosen during the training process (in order to ensure an ability to generalize but to avoid an excessive specialization in irrelevant details in the input image) [2]. The neural network with one hidden layer was trained on the basis of training set with the use of the backpropagation method. After the completed training process the neural network was ready to start working in the final recognition mode. In this mode every standardized input image processed by the neural network stimulated its output neuron, corresponding to the recognized class. The response of the neural network was interpreted and a decision about assigning the object into a recognized category/class (or about a lack of the recognition) was made.

## 4. INITIAL IMAGE PROCESSING AND RESULTS OF EXPERIMENTS

The described solution with the standardization of input images and the artificial neural network (the multilayer perceptron) as the features extractor and classifier simultaneously was applied in the carried out experiments. The aim was the recognition of objects from photos (showed below) which were divided into the training set (used only during training the neural network) and the test set (used to evaluate an effectiveness of this system). The test set includes photos of the same objects but these photos were taken from a little bit different perspective and with a different lighting. Both sets were increased by their modified versions, including objects scaled down (30% and 60% for the training set and 45% and 75% for the test set) and rotated ($72^o$, $144^o$, $216^o$, $288^o$ for the training set and $108^o$ and $252^o$ for the test set).

Fig. 5. Base images of the training set

Because the standardization of input images works properly only for binary images it is necessary to apply an initial processing of prepared photos sets. One of possible solutions of the initial images processing which allowed to achieve good results during carried out experiments is depicted in Figure 6. Following stages of the initial processing were applied before the standardization block.

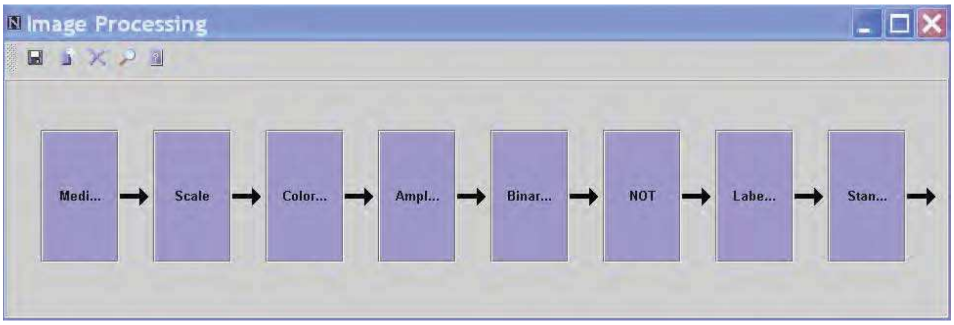



Fig. 6. Initial processing

First of all, an input image is passed by the median filter in order to remove the unfavorable noise (at a minimal change of the right image content which is an advantage of this filter). Next, the image is scaled down to the resolution 150x150 in order to reduce the computational complexity of further processing. Then the processed color image is changed into gray one and its contrast is enhanced (Color to Gray and Amplitude Rescaling blocks). The next step is the binarization (Binarize block). The NOT block ensures the proper representation of object pixels (value 1 for recognized objects, 0 for the background).

Finally, the image is labeled in order to distinguish and choose a correct object to recognize from other artifacts (Labeling block). Images processed this way are standardized (Standardization block) and passed to the neural network. An example of this processing, step by step, is depicted in Fig. 7.

The neural network with one hidden layer, containing neurons with logic activation functions, was trained on the basis of the training set and then the effectiveness of this trained network was checked with the use of the test set. During this process the number of hidden neurons was specified. It turned out that only 3 hidden neurons allowed the trained neural network to work properly.

Assuming that a signal of the output neuron, indicating the valid and reliable recognition of an object, should exceed signals on other outputs minimum 0.45 (a two-level interpretation) the effectiveness of the classification on the test set (which wasn't used in the training process) achieved 100%. Some results for particular input images were presented in the table 2. This recognition system worked very well even with very small objects with different locations or with objects which are not used in the training process or test set at all but belonged to one of recognized classes (for example, different screwdrivers marked in the table 2 with green descriptions).


Fig. 7. An example of the initial image processing

Table 2. Responses of the trained neural network for the test set

| **Image** | | **Winner** | **Neural network output value** | **Distance to maximum level of other outputs** |
|---|---|---|---|---|
|  | Big screwdriver | "screw driver" | 0.9413 | 0.9293 |
|  | Socket wrench rotated and scaled down 75% | "socket wrench" | 0.9790 | 0.8812 |
|  | Pliers rotated and scaled down 75% | "pliers" | 0.9903 | 0.9611 |
|  | Small screwdriver | "screw driver" | 0.9614 | 0.9593 |
|  | Socket wrench rotated and scaled down 45% | "socket wrench" | 0.8868 | 0.7293 |
|  | Pliers rotated and scaled down 45% | "pliers" | 0.9780 | 0.9341 |
|  | Another type of screwdriver | "screw driver" | 1.0 | 0.7404 |

## 5. REMARKS AND CONCLUSIONS

The examined system with the initial image processing is the solution where the neural network works as the features extractor and classifier simultaneously. No additional algorithm for an extraction of objects futures was needed. On the basis of carried out experiments it can be concluded that the applied neural network complited these tasks very well.

The system was insensitive to objects translation, scale and rotation in the input image (recognitions were correct thanks to applying the standardization of input images describe in this paper) and dealt with even with very small objects (scaled down 75%) in input images or with photos of objects taken at different lighting conditions and from a little bit different perspective. The system also recognized objects which belonged to one of the recognized classes but were not used in the training/test sets (marked with green in the table 2) which can confirm that the neural network was designed and trained properly because it had a generalization ability.

It seems to be obvoius that this recognition system works well for a higher frecuency noise or distortions. The input image is scaled to the matrix 15x15 before passing to the neural network. This scaling process reduces some details of the recognized objects (for example, as in Figure 7). It may be regarded as its advantage because of a distortions resistance. On the other hand, it may cause problems with a recognition of very similar objects but belonging to different classes. It is kind of trade off. However, it is always possible to increase the input matrix to allow the neural network to extract subtler features of objects at the expense of a higher computational complexity.

Generally, it is worth noting that the reconition of prepared test images (which were not used in the training process) worked perfectly in the designed system and it could perhaps prove correct in some real applications (for example, in a production process to recognize some parts automatically).

Certainly, it would be interesting to examine an influence of different kind of image distortions on the recognition process, especially that the neural network extracts features on its own. Authors are going to do this in future experiments.

## BIBLIOGRAPHY

[1] Jeatrakul P., Wong K.W., 2009. Comparing the Performance of Different Neural Networks for Binary Classification Problems. Eighth International Symposium on Natural Language Processing, pp. 111-115.

[2] Kotsiantis S.B., Zaharakis I.D., Pintelas P.E., 2007. Supervised Machine Learning: A Review of Classification Techniques. Informatica, vol. 31, pp. 159-190.

[3] Manjunath T.C., 2010. Detection of Shapes of Objects Using Sophisticated Image Processing Techniques. International Journal of Computer Science & Emerging Technologies, Volume 1, Issue 4, pp. 32-37.

[4] Masters T., 1996. Sieci neuronowe w praktyce. Programowanie w języku C++. Wydawnictwa Naukowo-Techniczne.

[5] Osowski S., 1996. Sieci neuronowe. Oficyna Wydawnicza Politechniki Warszawskiej.

[6] Osowski S., 1997. Sieci neuronowe w ujęciu algorytmicznym. Wydawnictwa Naukowo-Techniczne.
[7] Osowski S., 2006. Sieci neuronowe do przetwarzania informacji. Oficyna Wydawnicza Politechniki Warszawskiej.
[8] Zhang G.P., 2000. Neural Networks for Classification: A Survey. IEEE Transactions On Systems, Man, And Cybernetics - Part C: Applications And Reviews, vol. 30, No. 4, pp. 451-462.

## SYSTEM ROZPOZNAWANIA OBIEKTÓW Z WYKORZYSTANIEM SIECI NEURONOWYCH

Streszczenie

W pracy zaprojektowano i przetestowano, w zaimplementowanej przez autorów specjalistycznej aplikacji, system do rozpoznawania obiektów oparty na technice sieci neuronowych i zaawansowanym wstępnym przetwarzaniu obrazów. Praca obejmuje matematyczne podstawy standaryzacji przetwarzanych obrazów wejściowych, zastosowanej do osiągnięcia uniwersalnego i inwariantnego systemu, niezależnego od transformacji obiektów (przesunięcia, skalowania, rotacji). W pracy przedstawiono i omówiono wyniki przeprowadzonych eksperymentów z wykorzystaniem zdjęć zawierających testowe obiekty.

Słowa kluczowe: rozpoznawanie obiektów, przetwarzanie obrazów, sieci neuronowe