Recognition of alphanumeric characters using artificial neuron networks and MSER algorithm

Jan Matuszewski*[1],Marcin Zając[2]
[1]Military University of Technology, Faculty of Electronics, Institute of Radioelectronics, 2, gen. Sylwestra KaliskiegoSt.,00–908 Warsaw, Poland;[2]42 Baza Lotnictwa Szkolnego, Sadkow 9, 26-603 Radom, Poland

ABSTRACT

The article presents a method of recognizing alphanumeric characters located in the image, based on a previously created database of patterns using neural networks. For this purpose the convolutional networks were used, which independently search for features that allow to distinguish characters in the image. A larger number of convolution layers allows us to recognize a greater number of features and thus to increase the probability of correctly recognized characters. The main purpose of the paper is to present software that recognizes the alphanumeric characters in images and to investigate the impact of the size of this database on the program's speed and character recognition efficiency. This software can also be used in more complex structures, such as automatic translators or as a computer reader. The calculation of the first program that recognizes single character and the second program that reads all the text from the image have been made in the MATLAB environment. The paper describes the components of this software, such as the learning subsystem and the character recognition subsystem. The results of the program were presented in the form of screenshots showing the results of the learning process and character recognition process. The speed of the software and the effectiveness of recognizing alphanumeric characters using the artificial neural networks and maximally stable extremal regions (MSER) algorithm are presented in the table and figures. Attention was also paid to the impact of the size of the database used to learn the network on the speed of calculations and recognition efficiency.

Keywords: neural networks, alphanumeric characters recognition, MSER algorithm.

1. INTRODUCTION

In recent years, the Internet and publications published on it have been developed dynamically. However, the text written on a piece of paper is still an important way to transmit information. To enable the easy communication people with the computer it is also necessary to read information that are sent by people through electronic devices and processed in the understanding language for them. Electronic text recognition is used to solve this problem. The systems that deal with this task are known as Optical Character Recognition (OCR), or optical character discrimination systems [3, 22]. Such software usually uses neural networks that can learn from patterns and then read the human handwriting. The article presents the problem of recognizing alphanumeric characters using artificial neural networks [1, 7, 8, 19, 21]. A general scheme for images processing that detect alphanumeric characters was described and at each of its stages discussed. The results of calculations of a single character recognition, described in the article Artificial Neural Network Based Optical Character Recognition, were compared with the designed program based on the neural network consisting of 15 layers of neurons [24]. The impact of the size of the images database for network learning on the recognition results and learning time was also shown. The implementation of the algorithm for recognizing single alphanumeric characters in this application that allows reading text from photos is also presented.

[1]jan.matuszewski@wat.edu.pl; phone +48 261837571; fax +48 261 837 461
[2]marcin.zajac0246@gmail.com; phone +48 665811115

Radioelectronic Systems Conference 2019, edited by Piotr Kaniewski, Jan Matuszewski, Proc. of SPIE
Vol. 11442, 1144219 · © 2020 SPIE · CCC code: 0277-786X/20/$21 · doi: 10.1117/12.2565316

2. ALGORITHM OF OBJECTS DETECTION IN IMAGES

The proposed algorithm has the task to recognize the alphanumeric characters in the image being input. The scheme by which such an algorithm works is shown in Figure 1, [24].

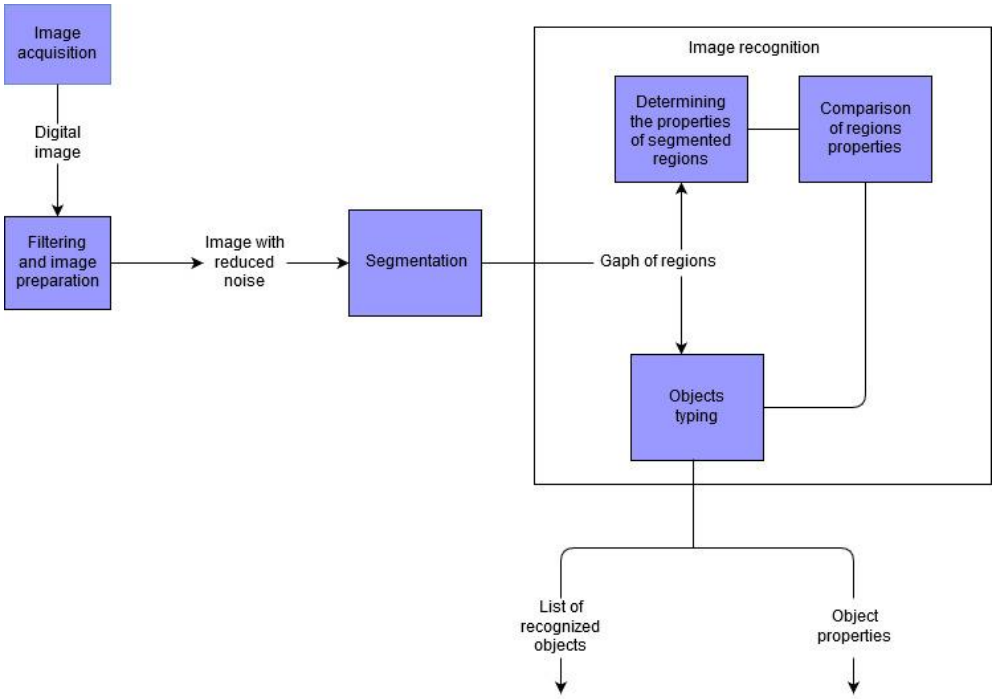

Figure 1. General scheme for detecting objects in images

The first stage of this algorithm is the image acquisition. It involves recording the image and presents it as a set of pixels. Such initial image processing enables further operations performed by the computer on it, then the image is subjected to filtration whose task it is to remove noise having a negative impact on the further image processing [10, 11].

The image processed in this way is most often subjected to binarization (Fig. 2a, b), [4]. Binarization transforms any image into a series of black pixels placed on a white background. Thus, all input images have the same dimension. In addition, it eliminates other effects such as contrast, sharpness, etc. It consists of two rounding pixel values, based on a specific threshold, this is: 0 - corresponding to black and 1 - corresponding to white colour.

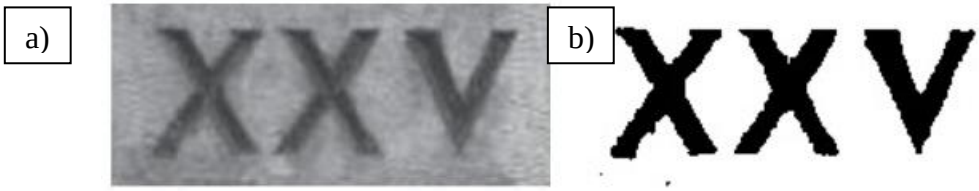

Figure 2. Image before binarization, b) Image after binarization

The next step of the algorithm is the image segmentation process [5, 20]. As a result, the fragments separated from the whole image are created, characterized by the uniformity in relation to colour, texture, etc. Characters usually strongly differ from the surroundings, which is why they are clearly exposed as a result of segmentation. Segmentation based on the gradient method is the most commonly way of segmentation used for reading characters.

Next the image is normalized in terms of size and sharpness for comparison with pre-loaded templates. In the next stage, the features are extracted and the individual parameters of the introduced characters are calculated. Based on them, the algorithm evaluates the entered character and assigns it to the most-fitting class. At the output of this program the recognized character is shown in the image.

3. SHORT CHARACTERISTICS OF USED NEURAL NETWORK

An artificial neural network (ANN) was used to recognize individual characters [15]. It is a mathematically defined structure and its model made in a software or hardware manner. The basic neuron model is shown in Figure 3.

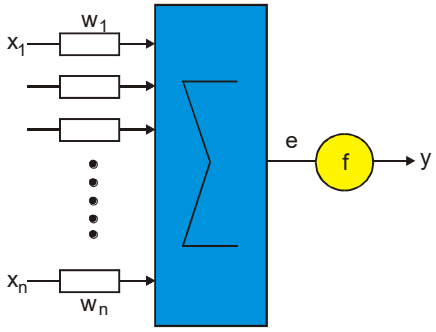

Figure 3. The basic neuron model

The symbols used in Figure 1 mean: $\mathbf{x}=(x_1, x_2,..., x_n)$ - vector of input values, $\mathbf{w}=(w_1, w_2,..., w_n)$ - vector of coefficient weights (synaptic weights), n – number of vector components, e - summary stimulation of the neuron

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

f- neuron activation function, y-signal value at the neuron output, $y=f(e)$.

The activation function is the basic element shaping the characteristics of the neuron. It determines not only the level of triggering the neuron, but also the range of value changes at its output. Within one neural network, its individual layers can be built of neurons with different types of activation functions. The most modern neural network models use the sigmoidal activation function (Fig. 4).

$$f(e) = \frac{1}{1+\exp(-\lambda e)} \tag{2}$$

It can be roughly defined as a continuous function with real values, whose domain is real numbers, which always has a positive derivative and whose range is limited. The basic advantage of the sigmoidal activation function is the easy determination of its derivative. Sometimes other sigmoidal functions are used, such as e.g. hyperbolic tangent (Fig. 5) and scaled arc tangent (Fig. 6), [8].

$$tgh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3}$$

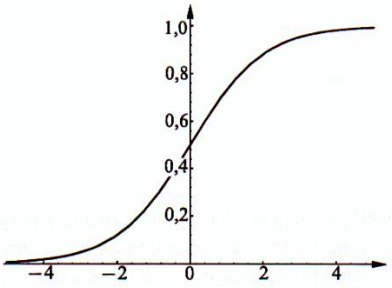

Figure 4. Sigmoid activation function

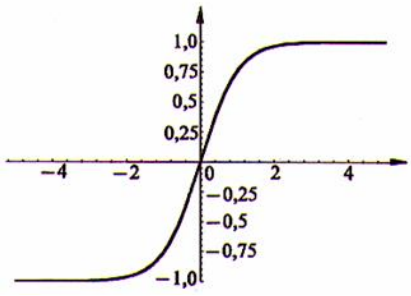

Figure 5. Hyperbolic tangent

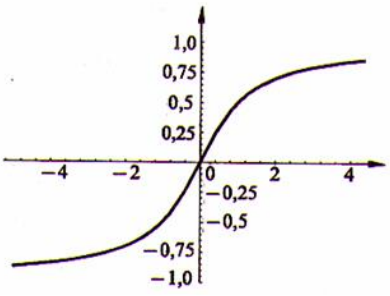

Figure 6. Scaled arcus tangent

It must be taken into account that neurons achieve full activation at a value of about 0.9 and a full zero state of approximately 0.1. The simplest unidirectional neural network consisting of the set of neurons logically located in the input and output layers, and possibly in hidden layers is depicted in Figure7. In the sense the input neurons are

hypothetical because they do not have their own inputs and do not perform any processing. Their activation is determined by giving a signal to the input of network. Designation that the network is unidirectional means that only one direction of signal flow in the network is possible, i.e. from input through the hidden layers to output. In such a network, the input signals to each layer, except for the first, come only from the output from the previous layer.

The neural network learned in this way minimizes the function criterion E, which takes into account the differences between the actual neuron responses y_i and the given d_i values.

$$E = \sum_{i=1}^{n} \left(y_i^{(k)} - d_i^{(k)} \right)^2 \tag{4}$$

The number of learning samples and the network learning time are rapidly increasing for more complex image shapes.

The purpose of learning the neural network is to determine the weighting factors $w_{ij}^{(1)}$ and $w_{ij}^{(2)}$ for all network layers in such a way that for the input signal $\boldsymbol{x}_i$ the output signal $\boldsymbol{y}_i$ is equal to the set value $\boldsymbol{d}_i$ with the assumed accuracy for all learning samples k. There can be only one input and output layer in the structure of a neural network, while hidden layers can be several. Typically, networks with more layers allow you to find deeper relationships and perform more accurate analysis, but this require more calculations and thus more time. Learning rules also have influence at the recognition pattern. Currently, more sophisticated and complex neuron networks are used to recognize a large number of images of different shapes, e.g. convolution neural networks, Fig. 8, [2, 6].

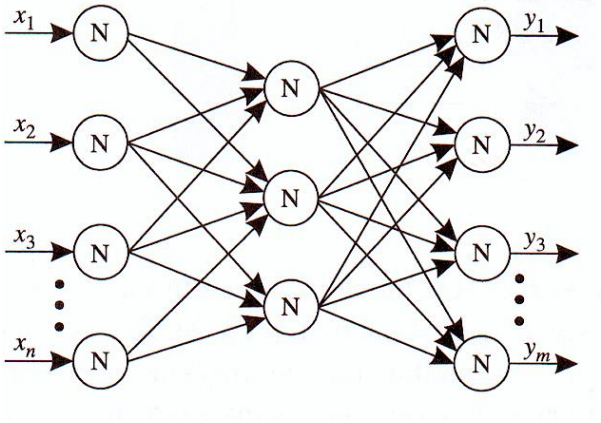

Figure 7. Unidirectional neural network model with one hidden layer

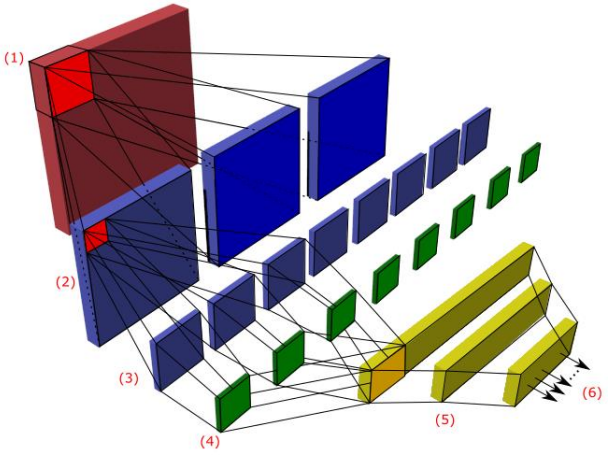

Figure 8. The structure of a convolutional neural network

A typical convolution neural network is the multilayer structure, except the first input layer (1), are convolution filters (2), threshold layers (3) and layers that change the image resolution (4)). The last layer (5) is usually several layers non-linear ANN with peer-to-peer connections and multiple outputs (6). The number of outputs depends on the number of images recognized by the network. The disadvantage of this type of network is the very long time of learning and need of millions of learning patterns. A special pattern/image generator is here needed. Also, the standard learning algorithm, i.e. the back propagation algorithm [17], requires modification and application of the analysis of second derivatives of error changes in relation to the weights, because the implementation of the traditional algorithm may be unstable and the learning period too long.

4. APPLICATION FOR RECOGNITION OF SINGLE CHARACTERS

4.1 Assumptions

The computer application for recognition the individual alphanumeric characters was developed under the following assumptions: the character read by the program is compared with the input image and that the recognition efficiency is determined by the software [23]. The set of data images is needed in order to learn this network. Based on them, the algorithm will be classify the entered characters. The images used for learning must be earlier properly formatted and next they can be input into the learning algorithm in the network. To do this, at the beginning should be changed the

images resolution to 32x32 pixels and if the data is in binary form, the values of all pixels must be transferred to three planes, corresponding to each of the RGB (Red, Green, Blue) components.

4.2 Program operation

On the basis of the above assumptions, a special algorithm was developed and then the computer program [13, 16, 24]. The *imresie* function was used to change the resolution of the images, and then the *Antialiasing* function to maintain the aspect ratio. Without this function, the algorithm would crop the image to specific sizes (in pixels), starting from the upper left corner, which would lose part of the image containing the necessary information needed to identify the character. With the help of this algorithm, a database was prepared for testing this character recognition application which contains 100 images for each from the 36 alphanumeric characters, and the database used to learn the network built in this way, which had 916 images for each character.

The designed application for recognition alphanumeric characters uses the artificial neural networks (ANN). The algorithm for its learning is built from the following 15 layers of neurons[24]:

1. *imageInputLayer*();
2. *conv1*;
3. *maxPooling2dLayer*();
4. *reluLayer*();
5. *convolution2dLayer*();
6. *reluLayer()*;
7. *averagePooling2dLayer*();
8. *convolution2dLayer*();
9. *reluLayer*();
10. *averagePooling2dLayer*();
11. *fc1*;
12. *reluLayer*();
13. *fc2*;
14. *softmaxLayer*();
15. *classificationLayer*().

Layer 1 is the input layer, into which the data are entered in the form of images with a resolution of 32x32 pixels. Layers 2, 5 and 8 are the convolutional layers, i.e. the connections between neurons in these networks are incomplete. Signals from neuron outputs are not forwarded to all neurons in the next layer. Layer 3 divides neurons into groups and selects the ones with the strongest response from them. Layers 4, 6, 9 and 12 are used to activate neurons by thresholding and setting zero values for all values smaller than it. Layers 7 and 10 divide the neurons into groups and average their values. Layers 11 and 13 are fully connected layers, i.e. each of their neurons is connected to each neuron of the next layer. Layer 14 (*softmaxLayer*) adapts the data from the input to the output layer. Layer 15 is the output layer which classifies images into 36 different classes corresponding to each alphanumeric character.

The program recognizes the alphanumeric characters from the entered image. The database containing 3600 images (100 for each character 0÷9 and A÷Z) was used to test the learning process in this network. The program classifies the testing images on the basis of information contained in the input database with the templates of individual characters. In addition, the program calculates the probability of correct characters recognition:

$$P_p = L_p / L_w \tag{5}$$

where L_p is the number of correctly recognized characters and L_w is the number of all test database images(3600).

4.3 Results of simulation program for recognizing alphanumeric characters

The simulation program was carried out for a different number of images used to learn the neural network. In each iteration, each of the 36 characters corresponded to: 10, 20, 30, 50, 100, 200, 400, 600 and 916 images used for learning. The test image database contained 3600 elements (100 for each character). The results of these calculations are presented in Table 1 and respectively in Figures 9 and 10, [24].

Table 1. Calculation results for the different number of images used to learn the network

The number of images used to learn the network corresponding to one character	**Learning time** [s]	**Probability of correct recognition of characters** [%]
10	9	24,11
20	18	45,22
30	29	61,44
50	52	89,97
100	106	92,92
200	197	96,06
400	391	97,31
600	629	97,56
916	1017	97,53

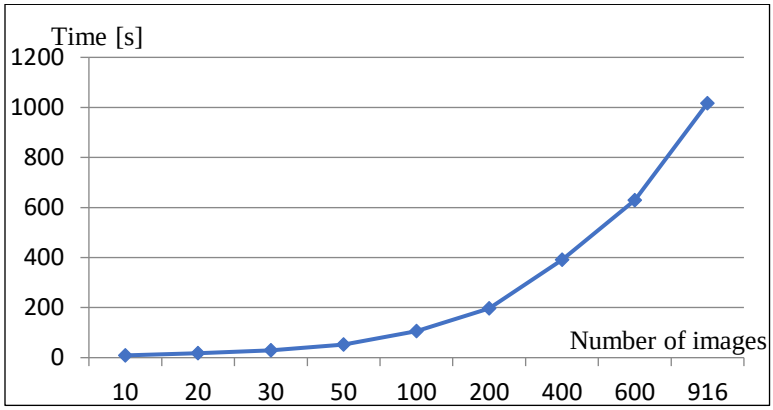

Figure 9. Dependence of neural network learning time on the number of images used for learning

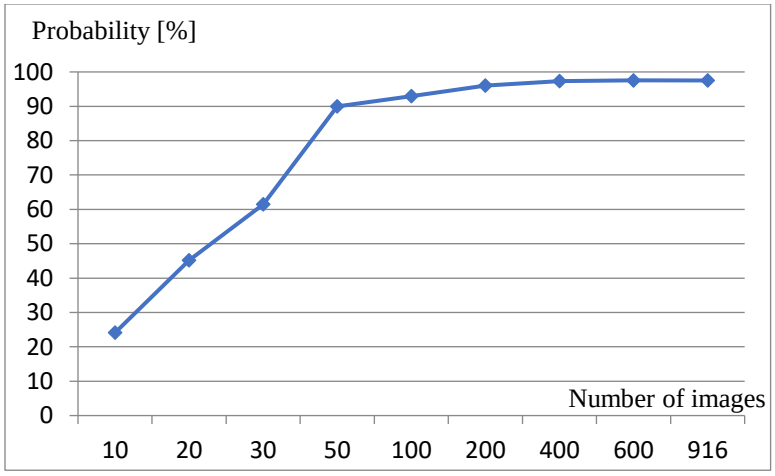

Figure 10. Dependence of probability of correct recognition of characters on the number of images used for learning

As a result of this calculations, it can be seen that the larger the image database used to learn the neural network gives the greater probability of the alphanumeric character recognition. However, when it reaches the value of over 97%, the increase in the number of database images used for learning only slightly increased recognition. If the number of learning images then the time to complete this process increases and for 916 images corresponding to each character it was over 1000 seconds.

5. APPLICATION FOR READING TEXT FROM IMAGES

5.1 Application assumptions

The application for reading text from images was developed on the assumption that the text read by the program is compared with the input image. It is used for images with a custom structure that contain indefinite or random areas. For example, it can automatically detect and recognize text from a captured video to notify the driver of a traffic sign. The division of text from a disordered environment helps in additional tasks, such as optical character recognition (OCR), [22]. Based on it the developed algorithm and the computer program detects a large number of potential areas of text and gradually removes those in which text is less likely.

5.2 Description of program operation

The MSER detector works well for searching text areas [14]. The program effectively detects text from the images because the subtitles have a consistent colour and high contrast compared to the surroundings. To find all regions with text in the image, the *detectMSERFeatures* function was used. It can be seen that in addition to the text, regions that do not contain text have also been detected. The result of the algorithm at this stage is shown in Figure 11, [24].

Although the MSER algorithm detects most of the text, it also selects areas in the image that are not the text. To remove non-text areas, it can be used the geometric properties of the text using the simple thresholds. In this example, a simple approach based on the geometric properties of the text was used to filter text areas.

There are several geometric properties that are good to distinguish between text and the environment [3, 12], including:

- proportion coefficient,
- centricity,
- enlargement,
- cohesion.

The *regionprops* function was used to determine these properties in the image, and then the text containing areas were limited. The result of the algorithm at this stage is depicted in Figure 12.

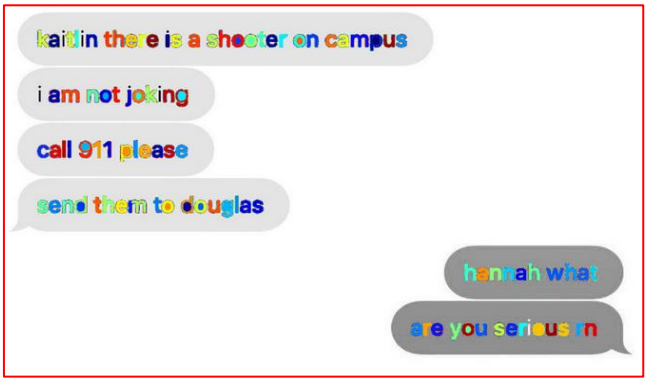

Figure 11. Detection of areas potentially containing text

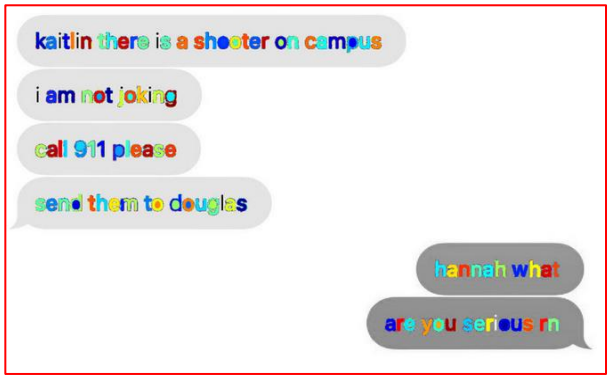

Figure 12. Eliminating areas that do not meet the geometric properties of the text

5.3 Area elimination based on the stroke width and merging the detected areas for final text

Stroke width is another common method used to distinguish text. It is a measure of the width of the curves and lines that make up the character. Typically, text regions have little variation in stroke width, while non-text areas have larger

differences in stroke. The distance transformation and binary thinning operations was used [12]. The result of using the stroke width to remove non-text areas is shown in one of the detected MSER regions in Figure 13.

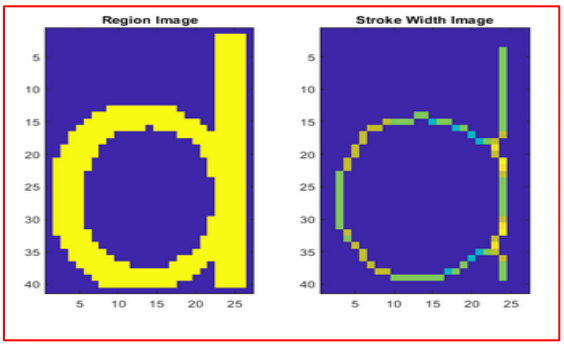

Figure 13. Image area and stroke width

Figure 13 shows how the stroke width has a small variance relative to most areas. This allows to specify it as a text region because the lines and curves that make up the region have similar widths, which is a property of the written text. To apply change of the stroke width to removing non-text areas, the threshold value must be set. The above algorithm must be carried out separately for each MSER area detected [18]. The result of the algorithm at this stage is shown in the figure 14, [24]. At this stage of the program, all detection regions consist of single text character. To use these results for recognition pattern tasks, such as OCR, individual text characters must be combined into words. This makes it possible to recognize the whole sentence in the image that contain more information than individual characters. In addition, these characters must be read in the correct order to preserve the meaning of the recognized words.

One way to combine individual areas of text into words is to find the adjacent areas of text and then create a bounding box around those areas. To find adjacent areas, expand the boundaries previously calculated using the *regionprops* function. This causes that the boundaries of adjacent text areas to overlapping, so that the detected areas form a chain of overlapping bounding boxes. The image after the frame expansion algorithm is shown in Figure 15, [24].

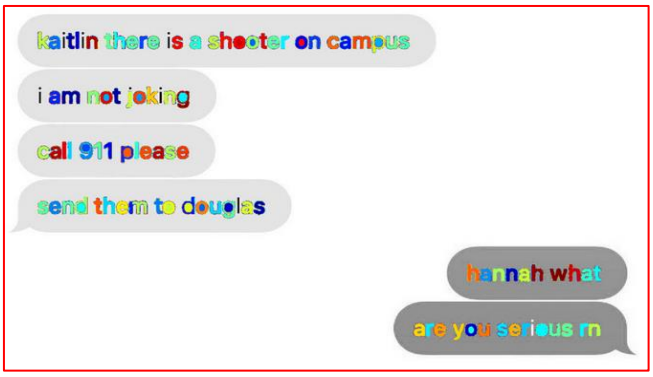

Figure 14. Output image after elimination of areas based on the stroke width

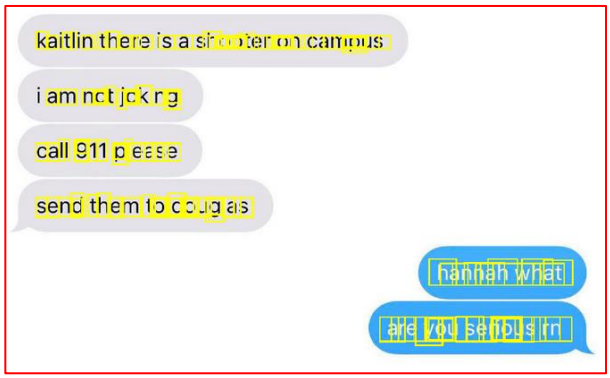

Figure 15. Image with the overlapping frames

From the resulting envelopes, it can be created frames around the whole words and combine the overlapping ones. At the end of the program the overlapping ratio between all envelope pairs is calculated. This determines the distance between all pairs of selected text areas, so the groups of adjacent text regions can be find with non-zero overlapping coefficients. After these calculation, it can combine areas that have the certain conditions. The *bboxOverlapRatio* function was used to calculate the overlappig factors. Based on it, the appropriate frames from the detected areas of the text were combined.

The output data *conncomp* are indexes for the connected text areas to which each bounding frame belongs. They are used to connect adjacent several bounding frames into one. By calculating the minimum and maximum of individual frames, each connected text component is created. Before displaying the final detection results the false text detection

should be excluded. Restriction frames that consist of only one text area are removed. This eliminates the individual areas that are unlikely to be actual text, because the text is usually found in groups (words and sentences). The result of the algorithm at this stage is shown in Figure 16. After detecting text areas, the text in each bounding box is recognized by the OCR function, Fig. 17.

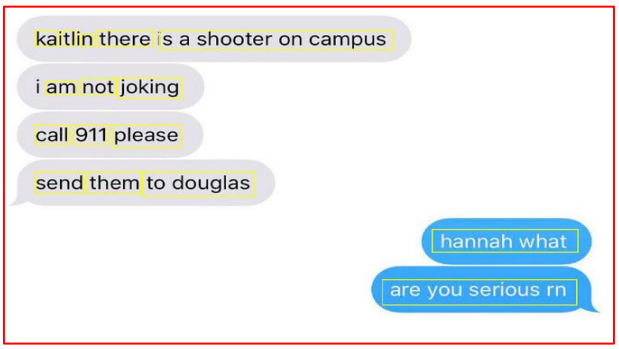

Figure 16. Image with detected text

```
ans =
    'Vt1in there is a shooter on campus
     am not joking
     call 911 please
     send
     them
     to Douglas
     hannah what
     are you serious rn

     '
```

Figure 17. Recognized text from the image

This program shows how to detect text on an image using the MSER detection function. First, should be find areas that potentially contain text. It then removes regions that contain no text using geometric measurements. The developed application does not have 100% efficiency, but correctly reads virtually all the text. In addition, it provides a good basis for further improvement and greater effectiveness.

6. CONCLUSIONS

The presented alphanumeric character recognition applications effectively recognize the characters in the images. The influence of the number of database images used to learn the neural network on recognition efficiency was also presented. The obtained probabilities of correct recognition were already above 97.3% for 400 images corresponding to one character. Attention should also be paid to the impact of the amount of data used to learn the neural networks in the learning time of this network. To optimize the work of programs based on the neural networks, it need to properly select the size of the database for the network learning stage to achieve the assumed recognition efficiency. For a number of images used at the learning stage, such a high degree of recognition efficiency is achieved that increasing the database significantly extends the learning process, but no longer brings commensurate benefits in increasing the recognition rate.

Based on carried out the simulation tests, it could be observed that, despite the huge database on the basis of which the characteristics of individual classes are determined, 100% recognition efficiency was not obtained, because the images that were subjected to the character recognition process were original and were not in the database used for teaching in exactly the same form. There will be always the minimal differences between these images and they will be recognized only on the basis of the characteristics similarity. As the recognition rate achieved over 97%, that means the algorithm extracted practically the maximum amount of information about a given character, so a further increase of learning images causes a disturbance of the "image" and the probability of correct recognition decreases slightly.

The described above the computer program used to read the text from the image made a small mistake only in one place, but taking into account the ratio of correctly read words to words read incorrectly shows very good recognition efficiency. The presented algorithm and program can be used in the field of alphanumeric characters recognition.

REFERENCES

[1] Chen, Huizhong, et al., “Robust Text Detection in Natural Images with Edge-Enhanced Maximally Stable Extremal Regions,” Image Processing (ICIP), 2011 18th IEEE International Conference on. IEEE (2011).
[2] 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).
[3] Gonzalez, Alvaro, et al., “Text location in complex images. Pattern Recognition (ICPR),” 21st International Conference on. IEEE (2012).
[4] http://analizaobrazu.x25.pl/articles/17 (6 Mart 2019).

[5] 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.
[6] Huang, Z.; Pan, Z. and Lei, B., “Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data,” Remote Sensing, 9, 907, (2017).
[7] Kirichenko L., Radivilova T. and Bulakh V., “Machine Learning in Classification Time Series with Fractal Properties Data,” vol.4, issue 1, 5, 1-13, 2019. DOI:10.3390/data4010005 (2019).
[8] 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., Radetskaya S. (eds) Lecture Notes in Computational Intelligence and Decision Making. ISDMCI 2019. Advances in Intelligent Systems and Computing, vol. 1020. Springer, Cham,701-711, https://doi.org/10.1007/978-3-030-26474-1_49 (2020).
[9] Kirichenko L., RadivilovaT. and Bulakh V., "Classification of Fractal Time Series Using Recurrence Plots," 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 719-724. DOI: 10.1109/INFOCOMMST.2018.8632010 (2018).
[10] Konatowski, S., “The development of nonlinear filtering algorithms,” Przeglad Elektrotechniczny, Vol. 86, Issue 9, 272-277 (2010).
[11] Konatowski, S. and Sosnowski, B., “Accuracy evaluation of the estimation process by selected non-linear filters,” Przeglad Elektrotechniczny, Vol. 87, Issuse 9A, 101-106 (2011).
[12] Li, Yao and Huchuan Lu, “Scene text detection via stroke width”, Pattern Recognition (ICPR), 21st International Conference on. IEEE (2012).
[13] Microsoft Software Developer Network (MSDN), Visual Studio IDE User’s Guide: https://msdn.microsoft.com/en-us/library/jj620919(v=vs.120).aspx (19 May 2015).
[14] Neumann, L. and Matas J., “Real-time scene text localization and recognition”, Computer Vision and Pattern Recognition (CVPR), IEEE Conference on. IEEE (2012).
[15] Osowski, S., [Sieci neuronowe do przetwarzania informacji], Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa (2006).
[16] Parrallel Neural Network Training with OpenCL: https://bib.irb.hr/datoteka/584308.MIPRO_2011_Nenad.pdf (27 October 2018).
[17] Petrov, N. and Jordanov, I., “Radar Emitter Signals Recognition and Classification with Feedforward Networks,” 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES2013,Procedia Computer Science, Vol. 22, 1192-1200, DOI: 10.1016/j.procs.2013.09.206 (2013).
[18] Pietkiewicz, T.,” Application of fusion of two classifiers based on principal component analysis method and time series comparison to recognize maritime objects upon FLIR images”, Proceedings of SPIE 11055, XII Conference on Reconnaissance and Electronic Warfare Systems, 110550Z, DOI: 10.1117/12.2524975 (2019).
[19] Pietkiewicz, T. and Sikorska-Łukasiewicz, K., ” Comparison of two classifiers based on neural networks and the DTW method of comparing time series to recognize maritime objects upon FLIR images”, Proceedings of SPIE 11055, XII Conference on Reconnaissance and Electronic Warfare Systems; 110550V, doi.org/10.1117/12.2524918 (2019).
[20] Prudyus, I. and Hryvachevskyi, A., “Image segmentation based on cluster analysis of multispectral monitoring data,” Modern Problems of Radio Engineering, Telecommunications and Computer Science, Proceedings of the 13th International Conference on TCSET 2016, 7452020, 226-229, DOI:10.1109/TCSET.2016.7452020 (2016).
[21] 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).
[22] Vivek Shrivastava and Navdeep Sharma, “Artificial Neural Network Based Optical Character Recognition”, Signal & Image Processing : An International Journal (SIPIJ), Vol .3, No.5 (2012).
[23] Wajszczyk, B., “Analysis of using a MicroBlaze processor for hardware implementation of algorithms for data processing in electronic recognition devices and systems based on the example of a XILINX FPGA system,” Proceedings of SPIE - The International Society for Optical Engineering, XII Conference on Reconnaissance and Electronic Warfare Systems, 110551F (27 March 2019), doi.org/10.1117/12.2525056 (2019).
[24] Zajac M., [Zastosowanie sztucznych sieci neuronowych do rozpoznawania znakow alfanumerycznych]. Praca dyplomowa, WAT, Warszawa (2019).