INSTYTUT BADAŃ SYSTEMOWYCH
POLSKIEJ AKADEMII NAUK

TECHNIKI INFORMACYJNE
TEORIA I ZASTOSOWANIA

Wybrane problemy
Tom 4 (16)

poprzednio

ANALIZA SYSTEMOWA W FINANSACH
I ZARZĄDZANIU

Pod redakcją
Andrzeja MYŚLIŃSKIEGO

Warszawa 2014

INSTYTUT BADAŃ SYSTEMOWYCH
POLSKIEJ AKADEMII NAUK

TECHNIKI INFORMACYJNE
TEORIA I ZASTOSOWANIA

Wybrane problemy
Tom 4 (16)

poprzednio

ANALIZA SYSTEMOWA W FINANSACH
I ZARZĄDZANIU

Pod redakcją
Andrzeja Myślińskiego

Warszawa 2014

Wykaz opiniodawców artykułów zamieszczonych
w niniejszym tomie:

Prof. Bernard De BAETS

Dr hab. Ewa BEDNARCZUK, prof. PAN

Dr hab. inż. Wiesław KRAJEWSKI, prof. PAN

Dr hab. inż. Andrzej MYŚLIŃSKI, prof. PAN

Dr inż. Jan W. OWSIŃSKI

Dr hab. Dominik ŚLĘZAK, prof. UW

Prof. dr hab. inż. Andrzej STRASZAK

Prof. dr hab. inż. Stanisław WALUKIEWICZ

ISBN 83-894-7555-3

CHARACTERS RECOGNITION BASED ON NETWORK OF COMPARATORS

Łukasz Sosnowski and Andrzej Pietruszka

Systems Research Institute, Polish Academy of Sciences,
Ph. D. Studies, Warsaw, Poland
and Dituel Sp. z.o.o.
l.sosnowski@dituel.pl
a.pietruszka@dituel.pl

Abstract. This paper presents the modern approach to the optical character recognition problem. The way to build the hierarchical classifier using network of comparators to solve OCR task is described. This paper is a continuation of previous research on compound object comparators and it includes an example of the practical use of this theory. The paper describes in detail the solution proposed and the result obtained.
Keywords: networks of comparators, compound objects resemblance, optical character recognition, classifiers

1 INTRODUCTION

Optical character recognition (OCR) [1] is the subject of research on pattern recognition, computer vision and generally artificial intelligence (AI) [2]. The main goal is to recognize characters in images, (e.g. scanned documents) and convert them into editable electronic form. The recognition is based on patterns, such as letters, numbers and certain special symbols, like commas, question marks, etc. The learning procedure is usually performed by showing examples of characters of all types of patterns [3]. These examples are used to construct certain of prototypes of each class of characters. Prototype objects provide knowledge consisting in the strength and effectiveness of recognition. The basic idea is to build a ranking of prototype objects in the context of recognizing characters [4]. The ranking requires the definition of the evaluation criteria on which it is to be based. In most cases one criterion is not enough, therefore a number of features are considered and might be of use for the final assessment of similarity. The decision on the recognition of the character is made on the basis of similarity to well-defined prototypes.

The resemblance works on the features originating from the processed image and the set of prototypes. Feature extraction is a procedure decomposing a character into feature vector which is easily comparable. There is a number of feature types and techniques of their acquisition. One is better for typewriting characters, the other for the handwriting ones. They are obtained by relatively simple image processing operations mostly based on granulation [5] and pixels position interpretation.

The OCR is generally a system consisting of two main modules: preprocessing image and a classifier. The first part covers the widely used image processing techniques [6], such as filtering, scaling, thresholding, etc. All these treatments ensure a given standard and a certain minimum quality of the input image. It allows performing segmentation of objects that provides the set of data (sub-objects) to the main decision process. The second part is a field of AI and there are a lot of algorithms that constitute an application. One of the well-known algorithms is a k-nearest neighbours algorithm giving satisfactory results.

In this article, the use of network of comparators to character recognition problem is suggested. A dedicated logical component called a comparator [7] is the basic element of the concept. It is responsible for examining the resemblance of a given feature between an input object and reference objects [8]. The comparator can be formally described as a function

$$C_B : A \to 2^{B\times[0,1]}, \tag{1}$$

where A is a set of input objects and B is a set of reference objects. Comparator outcome takes a form of weighted subsets of reference objects

$$C_B(a) = F(\{(b, g(\mu(a,b)) : b \in B\}), \tag{2}$$

where F is a function responsible for filtering partial results of a single comparator, e.g. min, max, top. Furthermore, $\mu(a,b)$ is a membership function of the fuzzy relation [9], which returns a similarity degree between $a \in A$ and $b \in B$, and $g(x)$ is an activation function which filters out results that are too weak. We put

$$g(x) = \begin{cases} 0 : x < p, \\ x : x \geq p, \end{cases} \tag{3}$$

where p denotes the lowest acceptable similarity. One may also introduce certain constraints, which make $\mu(a,b) = 0$ based on the so-called exception rules [7].

The approach is based on a network of mentioned comparators. This concept makes it possible to design a structured driven solution as well as a flat one. The network consists of layers. They include comparators,

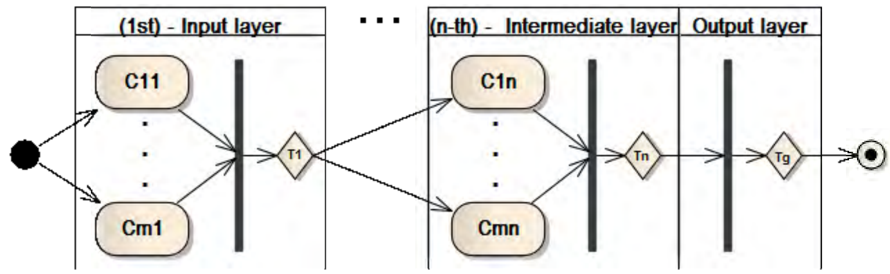

Fig. 1. General scheme of the network of comparators. This is not an UML activity diagram. It simply uses a similar notation: oval boxes represent comparators ($C_{11}...C_{mn}$), black vertical lines are aggregators, the one in the output layer is a global aggregator. Diamonds at the end of layers are translators [11] ($T_1...T_n$,T_g).

aggregators and translators [10]. There are two types of aggregators: local and global. The functionality of a local aggregator comes down to selecting the best results for a given layer based on partial results. The functionality of the second one is focused on the synthesis of results of individual layers in order to calculate the final result. The translator is a unit expressing the results of one layer by objects existing in another layer. The general scheme of the type of network in question is shown in Figure 1. Complete information of the construction and operation of the network of comparators is not the subject of this article. It has been well described is previous publication [10].

This article attempts to describe the way how the network of comparators can be used in character recognition and present the results achieved. The main focus is on the second part of the OCR system and the way to build a classifier [12] using a network of compound object comparators is described.

The paper is organized as follows: the first section provides introductory information about the context and background of the optical character recognition problem. The second section describes methods used to construct the solution. The third section presents the experiments, data used and results obtained. The subsequent section discusses the results and provides certain interpretations and suggestions. Criticism of the methods used is presented as well. The final part contains a brief summary.

2 METHODS USED

Similarity based reasoning is one of the approaches known in field of AI [13]. It assumes that there is a correlation between a similarity of two objects and that they belong to the same class or set of objects. Similarity expresses a *degree of identity*, which can be expressed by a question: to what degree object *a* resembles object *b*? Values of the similarity degree are often put into an interval [0,1], where *1* means that two objects are indiscernible. Properties of the similarity function depend on the domain of objects compared. Its values often depend on the context, and thus similarity is difficult to model by means of standard methods [14, 15]. A framework of compound object comparators [16] is used for modeling the resemblance of characters.

The designed network is made up of three layers representing particular contexts. The first layer covers a very general nature (rough) features. They can significantly reduce cardinality of the reference set. The second layer relates to in-depth analysis of the image [12], providing the final answer (decision). The last layer in classical way only contains an aggregator performing the synthesis of the previously obtained results. The scheme of the network of compound object comparators is shown in Figure 2. The

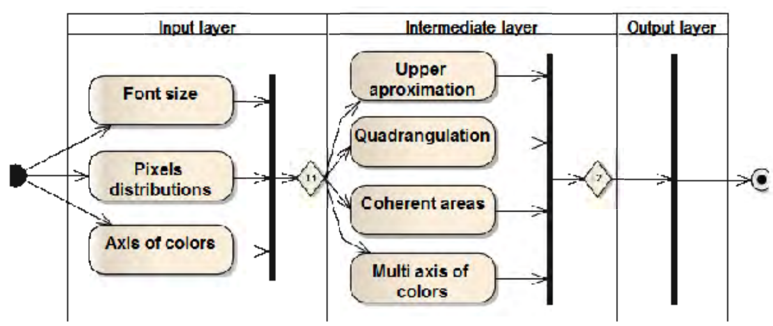

Fig. 2. The network of comparators for the OCR task. The input layer filters out reference objects by rough features. The local aggregator synthesizes partial results. The intermediate layer is responsible for accurate calculation of specific characteristics.

reference set consists of objects which are images of characters grouped by different sizes and font types. The cardinality of the reference set increase with the number of types of recognized characters, languages, fonts, sizes, etc. In practice, one of the main problems arises that there are have plenty of data and an efficient procedure to preselect some of them is necessary for further processing. The type I network is very similar to a hashing algorithm [17], which implements the said procedure.

Before the proper processing [6] is initiated by means of the designed network, the pre-processing of every object has to be performed. Consequently, a segmented string as a set of individual images is obtained. Each image represents one digit (as a single object). Each of these objects has to be converted then to a binary scale (only black & white). Eventually, each image is cut in a way that each edge of the newly created image is one pixel farther from the black edge of the font.

Each object has its dedicated representation for a particular comparator. The representation depends on operations needed to be performed during comparison. The network in question has three comparators in the input layer: font size, pixels distribution and axis of colors. Representations of objects used for them are the following: the height of the character in pixels, list of four quantities of the black pixel for each quarter of the image, two string patterns created from axis X and axis Y of the image (in the middle) and representing the pixel color changing. Figure 3 and 4 provide examples of the selected representation for certain comparators of the network.

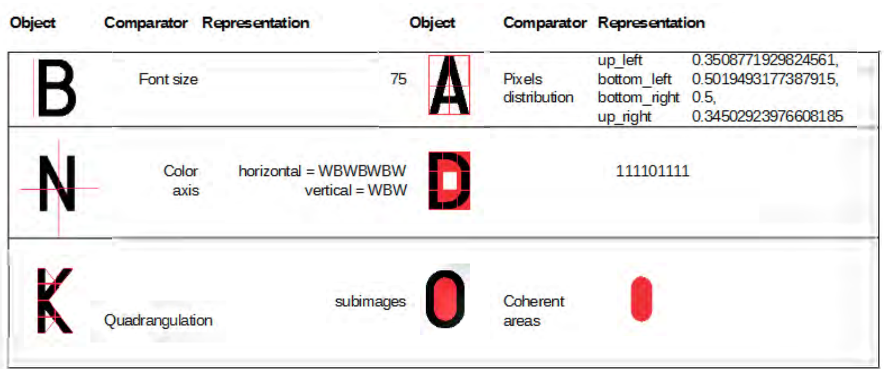

Fig. 3. Representations of selected objects used in solutions in question.

Each comparator computes similarity between the input object (character to recognize) and particular reference objects (images with the already assigned meanings, e.g. letter ”A”, digit ”4”, etc.). Three subsets of objects are selected in the input layer corresponding to the best partial results for a particular comparator. The further step is the synthesis [18] of results returned by all comparators from a particular layer. Aggregation is implemented by the arithmetic mean value of partial similarities. Results calculated by the aggregator feed the next layer, as a set of references. There are four comparators in the intermediate layer: Upper approximation, Quadrangulation, Coherent area, Multi axis of colors. The first comparator compares images originating from the granulation process [19]. The upper approximation is represented by granules activated if and only if there are black pixels inside. This reulsts in the comparison of images converted to a very low resolution ($m \times n$ granulation parameters) and represented by an array of $\{0,1\}$values (1 - activated, 0 - none).

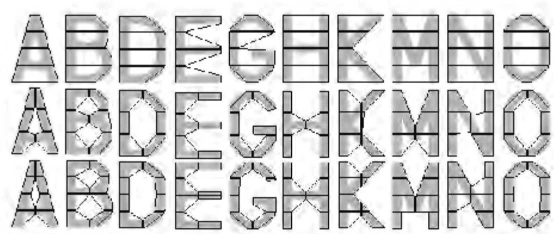

Fig. 4. Reference objects prepared for comparisons by a quadrangulation comparator with respect to their shapes (with granulation resolution parameters 1×4, 2×3, 2×4 displayed in consecutive rows).

Another comparator deals with comparison of geometric shapes resulting from the connection of extremes points of subsequent granules (see Figure 4). Subsequently, contours are taken and surface area of quadrangles is calculated. This value is required for the purposes of calculation of the final factor in form of a quotient of the surface area of quadrangle to the area of the whole image. The quadrangulation is performed by means of previously developed and published methods [21]. The third comparator is prepared to compare coherent areas within objects. Consequently, easier recognition of the character is supported, where the character contains a coherent area closed by the edge of the font, e.g. 0, 4, 6, 8, 9, A, B, D, O, P, R. This comparator not only detects the existence of such areas, but also indicates the most resembled areas in objects of reference. The last comparator in this layer (Multi axis of colors) compares string patterns created on the basis of color changing in particular lines of the image. There are two separate representations: one for horizontal lines and the other for vertical ones. The string patterns have been created analogically to the one used in the input layer, but now the focus is on each line (not only the middle one). Ones the pattern for particular lines is generated, only those which have changed (from the previous state) are obtained. Consequently, two lists of string patterns are obtained to compare.

In consequence, the calculated results of the subsequent comparator feed the output layer as input data of aggregator. The aggregator calculates an arithmetic mean of resemblance for particular pairs (the input object and the reference object) and the subsequent comparators in question. Finally, the ordered subset of reference objects with similarity assigned is obtained.

3 OBTAINED RESULTS

In this paper, data generated for the purposes of this research are used. A full set of digit objects (images) has been prepared and grouped by the following structure : size of the font (10, 14, 18, 24, 36, 48, 60); types of fonts (Times New Roman, Arial, Verdana, Courier). Additionally, there are objects representing letters occurring on license plates of cars (so there are no diacritics, punctuation, etc.). The letter set contains only one size and one font (the one used on the Polish car plates).

Generated images with characters and scanned images are used as input objects. The first set consists of character typed with different font type (Tahoma) than the one in the reference set. The scanned one has several sets of character types with various font types. Figure 5 shows the example of input sets.

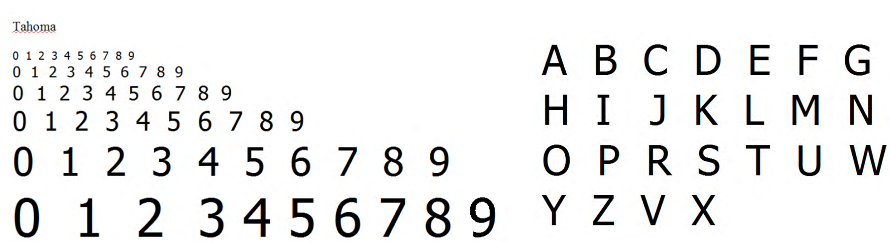

Fig. 5. Example of sets of input objects used for testing performance of methods presented.

Experiments employed measures dedicated for classifiers, e.g.: precision, recall and F1-score. Two types of measurements were performed for

Table 1. Results achieved for digits objects performed for the whole sets of input data with the same font size and different font types. The input characters (objects) were of various origin. Most of them were scanned, but some were computer generated.

Set	Precision	Recall	F1-score
Size 10	0.85	0.83	0.84
Size 14	0.92	0.88	0.90
Size 18	0.92	0.88	0.90
Size 24	0.93	0.80	0.86
Size 36	0.98	0.81	0.89
Size 48	0.98	0.74	0.83
ALL	0.93	0.82	0.87

each of the input objects sets. The first one was on the assessment of individual pairs for all input objects as classification results while the other evaluated the efficacy from the perspective of individual characters.

Three experiments were performed. The first one is a very basic one, only to make sure that the procedure is not inconsistent. It involves on inserting the reference set as an input one. If the processing procedure is correct, all the objects should match with similarity degree 1. This was the case here. The second experiment was based on digit characters with a structured reference set. Particular sizes of fonts were considered a separate set of input objects. Each of them contains 40 objects divided into 4 groups. Three of them represent scanned characters (300 DPI) with three different font types. The last group included the same font as the one from the previous group but the objects were generated (not scanned). Finally, a test with all input objects (all sizes and all types) against full reference set was performed. The results achieved are shown in Table 1. A different point of view is presented in Table 2. It contains results achieved for individuals characters (in this case digits). The table shows which characters have bigger problems in recognizing than others.

The third experiment was performed on characters from car plates (letters and digits). The reference set did not contain any structure and input objects (images) had only one font type (different than the one in reference objects). Types of calculate characteristics were the same as in the second experiment. Results are shown in Tables 3 and 4. The summary of individual character results as well as results for sets are grouped on the chart shown in Figure 6.

Table 2. Results achieved for digits objects performed for individual characters for all font types and all processed font sizes. The input characters (objects) were of various origin. Most of them were scanned, but some were computer generated.

Character	Precision	Recall	F1-score
0	0.96	1.00	0.98
1	1.00	1.00	1.00
2	0.89	1.00	0.94
3	0.95	0.83	0.89
4	0.96	0.96	0.96
5	0.88	0.96	0.92
6	0.76	0.92	0.83
7	1.00	1.00	1.00
8	1.00	0.96	0.98
9	0.94	0.67	0.78
ALL	0.93	0.82	0.87

Table 3. Results achieved for letters objects performed for the whole sets of input data with the same font size and different font type (Tahoma). The input characters were computer generated.

Set	Precision	Recall	F1-score
Size 10	0.68	0.68	0.68
Size 14	0.68	0.68	0.68
Size 18	0.76	0.76	0.76
Size 24	0.80	0.80	0.80
Size 36	0.84	0.84	0.84
Size 48	0.77	0.80	0.78
ALL	0.75	0.76	0.76

4 DISCUSSION

Results presented for digits prove that networks of comparators provide an appropriate tool for solving problems such as character recognition. Value of the F1-score factor stood at 0.87, which is a very good result. Considering individual digits it is clear that many characters achieved the 1.0 value and only three of them have the value of less than 0.9. The result table shows that there is a problem with the recognition of digit ”6”. Probably there is a need to construct additional comparators in order to distinguish this object from other characters.

The results for letters show that the well-constructed reference set is one of the most important things in such solutions presented. In this case there is only one type of fonts collected and only one size. Additionally the font type is unusual and different than all others processed. Neverthless, results are satisfactory and give hope of achieving even better ones with a more complete set of reference and perhaps additional comparators. Results for

Table 4. Results achieved for letters objects performed for individual characters. The input characters were computer generated.

Character	Precision	Recall	F1-score	Character	Precision	Recall	F1-score
A	1.00	1.00	1.00	N	0.55	1.00	0.71
B	1.00	1.00	1.00	O	0.50	1.00	0.67
C	0.75	1.00	0.86	P	1.00	1.00	1.00
D	1.00	0.33	0.50	R	1.00	1.00	1.00
E	0.67	0.33	0.44	S	0.67	0.67	0.67
F	1.00	1.00	1.00	T	1.00	1.00	1.00
G	0.86	1.00	0.92	U	1.00	0.83	0.91
H	1.00	1.00	1.00	V	0.60	0.50	0.55
I	0.00	0.00	0.00	W	0.00	0.00	0.00
J	0.80	0.67	0.72	X	0.67	0.33	0.44
K	0.80	0.67	0.72	Y	0.67	1.00	0.80
L	1.00	1.00	1.00	Z	0.38	1.00	0.55
M	0.67	0.67	0.67	ALL	0.75	0.76	0.76

individual characters show that there are a lot letters with the highest F1-score value (1.0). This means that in the case of these letters no mistake was made. On the other hand, it can be noted that there is a considerable problem with the ”I” and ”W” letter. They did not match correctly even once. The analysis shows that the ”W” letter always matches ”M” or ”N”. This can be eliminated by introducing a new font type into the reference set and probably an additional comparator to handle the middle part of the upper side of the image.

5 CONCLUSIONS

The paper presented an approach to the problem of character recognition, which is based on the network of comparators. Implementation of the network of comparators is quite easy and intuitive. Domain knowledge, intended for a given feature or for the whole processing can be easily injected into the solution presented. Experiments conducted and results obtained show that this method is efficient for these kinds of problems.

The results obtained in this study can be further optimized and developed. There is still room for increasing efficiency of the model and extending functionality of the solution. The problem of approximation of vague concepts, such as character recognition, attracts many researchers from the AI domain. The approach of thus paper is a contribution to this field. One of the shortcomings of this approach is quite a laborious process of defining the features for the network of comparators (NoC) and qualitative rela-

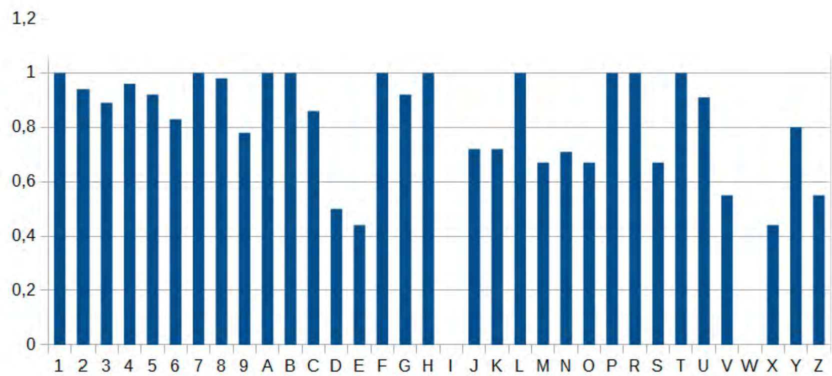

Fig. 6. Summary chart for all results expressed by F1-score values.

tions between the features (measures of similarity). However, the effort is rewarded by good performance of the classification.

References

1. A. Tormási and L. T. Kóczy (2013) Fuzzy-based multi-stroke character recognizer, In: *FedCSIS*, pp. 671–674.
2. S. O. Kuznetsov, D. P. Mandal, M. K. Kundu, and S. K. Pal, Eds. (2011) *Pattern Recognition and Machine Intelligence - 4th International Conference, PReMI 2011, Moscow, Russia, June 27 - July 1, 2011. Proceedings*, ser. Lecture Notes in Computer Science, vol. 6744. Springer.
3. T. M. Mitchell (1997) *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc.
4. N. Arica and F. T. Yarman-Vural, “An overview of character recognition focused on off-line handwriting,” *Trans. Sys. Man Cyber Part C*, vol. 31, no. 2, pp. 216–233, May 2001. [Online]. Available: http://dx.doi.org/10.1109/5326.941845
5. J. Stepaniuk (2008) *Rough-Granular Computing in Knowledge Discovery and Data Mining*. Ser. Studies in Computational Intelligence. Springer, vol. 152.
6. J. Russ (2011) *The Image Processing Handbook, Sixth Edition*. Taylor & Francis. [Online]. Available: http://books.google.pl/books?id=gxXXRJWfEsoC
7. D. Ślęzak and Ł. Sosnowski (2010) SQL-based Compound Object Comparators: A Case Study of Images Stored in ICE. In: *Proc. of FGIT-ASEA 2010*, ser. Communications in Computer and Information Science, vol. 117, pp. 303–316.
8. L. Sosnowski and D. Ślęzak (2013) How to design a network of comparators. In: *Brain and Health Informatics*, pp. 389–398.
9. J. Kacprzyk (2012) *Multistage Fuzzy Control: A Model-based Approach to Fuzzy Control and Decision Making*. John Wiley & Sons, Limited.
10. L. Sosnowski and D. Ślęzak (2013) Networks of compound object comparators. In: *FUZZ-IEEE*, pp. 1–8.
11. M. S. Szczuka, L. Sosnowski, A. Krasuski, and K. Krenski (2014) Using domain knowledge in initial stages of kdd: Optimization of compound object processing. In: *Fundam. Inform.*, vol. 129, no. 4, pp. 341–364.

12. M. Wozniak, M. Graña, and E. Corchado (2014) A survey of multiple classifier systems as hybrid systems. *Inf. Fusion*, vol. 16, pp. 3–17, [Online]. Available: http://dx.doi.org/10.1016/j.inffus.2013.04.006
13. L. Sosnowski, A. Pietruszka, A. Krasuski, and A. Janusz (2014) A resemblance based approach for recognition of risks at the fire ground. In: *Active Media Technology*, to appear.
14. A. Tversky and E. Shafir (2004) *Preference, Belief, and Similarity: Selected Writings*. Ser. Bradford books. MIT Press, [Online]. Available: http://books.google.pl/books?id=OCXAgA3sigIC
15. A. Janusz (2014) Algorithms for similarity relation learning from high dimensional data. *Transactions on Rough Sets XVII*, vol. 17, pp. 174–292.
16. Ł. Sosnowski (2012) Identification of compound objects using comparators. In: *Information Technology Theory and Application*, A. Myśliński, Ed., IBS PAN, vol. 2, pp. 114–135, in Polish.
17. N. Sundaram, A. Turmukhametova, N. Satish, T. Mostak, P. Indyk, S. Madden, and P. Dubey (2013) Streaming Similarity Search over One Billion Tweets Using Parallel Locality-Sensitive Hashing, *PVLDB*, vol. 6, no. 14, pp. 1930–1941.
18. L. Sosnowski (2014) Applications of comparators in data processing systems. *Technical Transactions, Automatic Control*, to appear.
19. W. Pedrycz, A. Skowron, and V. Kreinovich (2008) *Handbook of Granular Computing*. New York, NY, USA: Wiley-Interscience.

ROZPOZNAWANIE ZNAKÓW W OPARCIU O SIECI KOMPARATORÓW

Streszczenie. Niniejszy artykuł przedstawia nowoczesne podejście do problemu rozpoznawania znaków. Opisujemy w nim jak budować hierarchiczne klasyfikatory przy użyciu sieci komparatorów do rozwiązania zadania OCR. Ten artykuł jest kontynuacją poprzednich badań dotyczących komparatorów obiektów złożonych oraz zawiera przykłady praktycznego zasosowania tej teorii. Artykuł opisuje szczegółowo zaproponowane rozwiązanie oraz osiągnięte wyniki.

Słowa kluczowe: sieci komparatorów, podobieństwo złożonych obiektów, optyczne rozpoznawanie znaków, klasyfikatory

ISBN 83-894-7555-3