Application of Methods of Machine Learning for the Recognition of Mathematical Expressions

Oleh Veres[1][0000-0001-9149-4752], Ihor Rishnyak[2][0000-0001-5727-3438],
Halyna Rishniak[3][0000-0003-0976-0818]

[1,2,,3] Lviv Polytechnic National University, Bandery str., 12, Lviv, Ukraine, 79013
Oleh.M.Veres@lpnu.ua[1], Ihor.V.Rishnyak@lpnu.ua[2],
Halyna.M.Rishniak@lpnu.ua[3]

Abstract. The article describes the study of the peculiarities of presentation of mathematical methods, as well as methods and algorithms for their recognition. The possibility of simultaneous execution of structural analysis and character classification is investigated. The process of classification of the symbols and construction of the corresponding system, based on methods of machine learning, is described. For the initial initialization of the symbol classification process, a segmented binary image passes a "rough" classification by the Bayesian Network. Classification using contexts is processed by artificial Neural Networks. The system being developed is a multi-classifier. Five different classifiers work to get the optimal result.

Keywords: classification, classifier, symbol, structure, mathematical expression, machine learning, Bayesian Inference, Neural Networks.

1 Introduction

Computer vision is the theory and technology of developing systems that can find, track, classify and identify objects by extracting data from images and analyzing received information [1].

The purpose of Computer Vision and Pattern Recognition (MVPR) is to develop useful applications, especially through the use of processing and analysis of digital images. Computer vision is used to recognize objects, video analytics, description of image and video content, gesture recognition and handwriting, as well as intelligent image processing [2-5]. Statistical data uses statistical methods and uses models that are constructed using geometry, physics and theory of learning.

Today, computer vision is at the peak of its development. The speed of modern digital devices and the possibility of parallel computing provide the ability to implement many algorithms for working with digital image libraries.

2 Analysis of recent research and publications

Optical character recognition (OCR) is a mechanical or electronic transfer of handwritten, typewritten or printed text into sequences of codes used for presentation in a text editor [1]. Recognition is widely used for converting books and documents into electronic form, automating accounting systems in business or publishing text on the Internet. Optical recognition of images containing text is a widely studied problem at the interface between the field of artificial intelligence and computer vision.

Modern hardware and software systems allow automate large volumes of data into a computer, using, for example, a network scanner and parallel text recognition on multiple computers simultaneously. The most popular OCR systems are ABBYY FineReader, SimpleOCR, FreeOCR, Microsoft Office Document Imaging, and more.

The most difficult problems associated with the recognition of handwritten and printed characters are a variety of forms and ways of representing characters; distorting character images; variations in the size and scale of symbols (Tabl. 1).

Table 1. Comparison OCR systems.

OCR Systems	Forms and Ways	Distorting character	Variations in the Size and Scale
FineReader	Yes	No	Yes
SimpleOCR	Yes	No	Yes
FreeOCR	Yes	No	No
MODI	No	Yes	Yes
OCRFeeder	Yes	Yes	No

2.1 Overview of methods for recognizing mathematical expressions.

Some methods for recognizing expressions are based only on spatial measurements such as baselines [6, 7]. Other methods use rules-based systems and analyze the expression for its interpretation [8]. In several algorithms, knowledge of mathematical symbols and operators and their spatial properties are taken into account [9-16].

In Zanibby's works, the baselines that are in expressions [6] are analyzed. In particular, the dominant baseline is considered, which is the line on which the expression will be written and, for example, the embedded baselines that correspond to the indices. During the first step, a tree is constructed based on these baselines. Knowledge about the properties of mathematical notation for some tree transformations is also used. The resulting tree represents the content of the equation. In further research, Zanibby improved the recognition of indices and indexes using fuzzy regions [7]. This was motivated by the fact that most ambiguities in handwritten mathematical expressions refer to variants of index / line and line / upper index. In addition, the use of fuzzy logic makes it possible to return a ranked list of interpretations.

Tapia and Rojas first receive baselines and recursively build a mini-tree, in which each node is a symbol [6, 8]. In addition to baseline analysis, using a graph to represent the expression, they construct a minimal bias tree. Then a syntactic and semantic analysis is performed, using rules based on the features of the operator. Suzuki uses a network of virtual links [9]. Ray and Kim presented a method for conducting an effective search for structural analysis recognition [10]. Miller and Viola retain ambiguity during the character recognition stage [11]. They then calculate the probability that each character belongs to a certain class (small letter, number, binary operator, etc.), as well as the probability of being an index, upper index or linear expression, according to character recognition and some location properties. Chen performs both recognition and understanding of the formula [12]. Aval tried to simultaneously optimize segmentation and character recognition and structure for handwritten expressions [13]. Wang and Fore do not use any information about the character. According to the relative height of the two characters, they build the distribution of probabilities for bindings (index / string / upper index) between the symbols according to their relative vertical arrangement [14]. They also investigated the segmentation of manuscript forms based on human visual perception of a mathematical expression. Ali for the correct recognition of indexes and add-ins uses normalized bounding rectangles as the main feature of a character [15, 16]. A virtual remote element is added before the interconnection is recognized. They proved that with normalized restrictive rectangles, along with the special processing of the wrong characters, they can effectively recognize the connection using the Bayesian classifier.

3 Features of recognition of mathematical expressions

Available OCR systems are high-quality products in their field of application. However, the specificity of recognizing mathematical expressions requires more specialized software.

Character recognition is performed by classical methods of OCR, for example, using methods of reference vectors, coincidence with patterns. The analysis of the structure is mainly carried out with the help of geometric considerations, which are grounded on implicit rules or grammatical rules. Uncertainty in mathematical expressions, especially in manuscripts, is generally accepted. This may be uncertainty about the meaning of a symbol or structure. Despite the fact that artificial intelligence is used in the recognition of a structure with fuzzy logic or search algorithms, machine learning is not sufficiently used in the analysis of the structure. The development of science generates new mathematical notation. They may not be identified and break the structural analysis. There is a need for the ground-based application of machine learning methods for recognizing mathematical expressions.

The purpose of the study is to develop an intelligent recognition system for mathematical expressions based on machine learning, where character classification and structure analysis are separate tasks. An intelligent intelligence engineer should recognize mathematical expressions in two dimensional binary images and submit them in Latex format.

Mathematical expressions can be presented in a format that a person reads easily, or as a two-dimensional graph. They can also be submitted for use by computers. The presentation may be different from the reverse Polish record used in pocket calculators of the 1980s, to tree-like structures in some symbolic computing systems.

A mathematical expression is not just random symbols. They have a well-organized structure that is subject to the rules of the system of mathematical notation. The arrangement of two symbols relative to each other has a certain content.

The usual order of writing of reading for mathematical expressions is left-to-right. Therefore, understanding the mathematical expression is not completely two-dimensional. Interlinked characters are usually located next to each other. However, reading a mathematical expression is not straightforward, since different characters are usually read in different ways.

The main differences between plain text and mathematical expressions of the field are that mathematical formulas use many more symbols and have many types of connections. This justifies two main problems in the recognition of mathematical expressions, namely: the number of types of symbols and the context of their application; types of spatial relationships (upper index (or upper right), lower index (or lower right), in the same line, top, bottom, inside).

Mathematical expressions, as a rule, can be considered as embedded structures, especially because of the presence and properties of spatial relationships. Since the formulas are generally written in a line called the baseline, the embedded structure implies the presence of embedded baselines.

The most intuitive way to represent mathematical expressions is graphical. It is a printed or handwritten two-dimensional structure with symbols of various sizes and positions. This way of presenting is user friendly for reading and understanding the expression. There are other forms and are suitable for entering expressions into a computer. Simple formulas can be written on one line. To fix the logical content of the expression, it can be represented as a tree. There is also MathML - XML format. It captures the embedded properties of the mathematical expressions: Presentation MathML focuses on the spatial representation of the expression; Content MathML is a text translation of a tree view.

Recognition of mathematical formulas is a task where an image representing a mathematical expression is interpreted by the computer so that it can be stored, interpreted and reused. Recognition of mathematical formulas consists of two tasks (see Fig.1):

- *character recognition*: each pixel of the foreground in the image fills the character and each symbol is in the expression and transmits some information;
- *structure recognition*: the two-dimensional layout of the expression is subject to some rules and each scheme corresponds to a certain value.

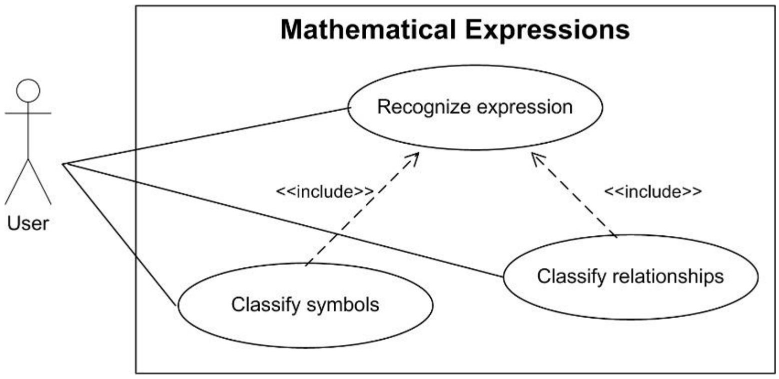


Fig. 1. Use Case Diagram of System

Character recognition is a procedure by which each symbol is recognized and classified. This is a difficult task because of the large number of sim-wolves. There is no dictionary, as for the recognition of the text. One and the same sim-wave may appear in different contexts, and it is important to distinguish, for example, symbol summation Σ and the Greek letter Σ. Some different characters have the same form, for example, p and P. Even more problems arise when it comes to hand-written expressions. To solve the problem of character recognition, neural networks or the methods of reference vectors are used.

The complexity of the task of **recognizing the structure** depends on the level of interpretation. Recognition of the expression structure is a collection of location analysis and interpretation of the representations of symbols and interconnections.

The structure of mathematical formulas looks quite simply to execute its recognition without the symbol value. There are several reasons for recognizing sim-wolves before recognizing the structure. First, the value of the characters is a huge limitation to the possible structure. However, symbols do not completely determine the location. The rules of their association are well structured. For example, the top index will never be found under its parent symbol. It is always located in the upper right corner. The main component in the coupling of the structure is not the characters, but their positions and sizes. The range of symbols and rules used to write mathematical expressions is not fixed. Common symbols and structural rules are just a subset, perhaps an infinite number, since characters and their new meanings can be invented at any time.

Spatial links between symbols are well defined, in limited quantities (index, upper index, etc.), but relationships can appear in a context in which they usually do not occur. Consequently, the result of the communication recognition should not affect the value or class of the symbol.

Character identification is not necessarily required for the recognition of the structure. Characters can be classified using only their restrictive rectangle and context.

The range of symbols and rules used to write mathematical expressions is not fixed. *The symbol context* is the information about the symbol itself (for example, a restrictive rectangle, a symbol class), as well as its parent and child symbols. Important features are their relative size and position, as well as the relationship between them. The expression from the arrangement of characters will be recognized. To do this, the symbols will be reduced to their bounding rectangles.

4 Methods and Technologies for the Recognition of Mathematical Expressions

Mathematical expressions with limited complexity will be concentrated on, namely:

- *zero order*: this is only a one-dimensional sequence of characters; it does not contain indices, upper indices, etc., for example, $a + b - N; \sum a \times \phi;$
- *first order*: an expression that contains one level of nested structures; when the indices and upper indices are expressions of zero order, for example, $a_p + b^{i+1}$; $\sum_{i=0}^{N} a_{i+1} + N;$
- *n*-th order: expressions in which the embedded structures have order $n - 1$.

The expressions of the zero order look simpler, because they are one-dimensional, therefore, it is a typical OCR task. Only spatial functions will be considered, so the amount of information in the expression of zero order is small. First-order expressions can be quite complex, since it is necessary, for example, to identify an expression as an index. We define the intermediate order. Expression of the 0.5 order is an expression in which the embedded expressions are separate characters. Example, $a_b + c_d^e$ this is an expression of 0.5 order, whereas a_{b+c} - no.

The input format is the image of a *handwritten mathematical expression*. It is necessary to develop a method for recognizing expressions of 0.5 order, as well as simple expressions of 1 and 1.5 orders, in order to check the reliability and ability of the system to adapt to more complex situations.

It is not only needed to recognize the structure, but also try to find a character class using this structure. A high order means a lot of context that should simplify the classification. However, when the expressions become more complex, the recognition of the structure is also complicated. It is important to make a compromise between problems that arise from the complexity and necessity of a context.

During the recognition, the format of the data changes. At the input, binary image is obtained, and the output must get the interpretation of the expression. As a system input, a binary (binary) image is selected. The segmentation algorithm reads the image and obtains related components. From the found components only restrictive rectangles are stored. This is the easiest way to present a layout of the expression. Each element in the list of bounding rectangles has the form $x_{min}, x_{max}, y_{min}, y_{max}$ - this is the left, right, upper and lower limits of each rectangle.

The list of bounding rectangles is used to create a representation of the expression, which is a list of characters. The essence of the "symbol" represents a symbol without context. It is created from the coordinates of the restrictive rectangular. The essence contains information about the size and position of the character. The essence of the "context" is a symbol with its context. It can be associated with other characters, such as a parent symbol or its child elements. It also contains a connection that has a character with its parent element, and the probability value for the symbol class and relationships. The essence of the "expression" consists of a list of characters. Creating an expression creates the "character" object for each bounding rectangle in the list and creates a "context" for each character. All contexts are stored in a list that is an "expression".

To find the relationship between characters and define character classes, the *classification* is performed. This can be seen as bundling characters together and adding information to an existing structure. A tree is created from the initial list. Each node corresponds to the context. Information contained in the node: the corresponding character; connection with the parent element; regions where child elements should be found; the distribution of probability values over the possible classes of characters for each classifier; the distribution of probability values for possible relationships with the parent element. Each subsidiary node corresponds to the child element of the symbol represented by that node. There is also a feedback link to access the parent's character.

To classify characters, several technologies of machine learning are used. Bayesian Inference carries a rough classification. Classification with the use of context is handled by artificial Neural Networks. Different classifiers work separately but are used together. They are tied to return the optimal result.

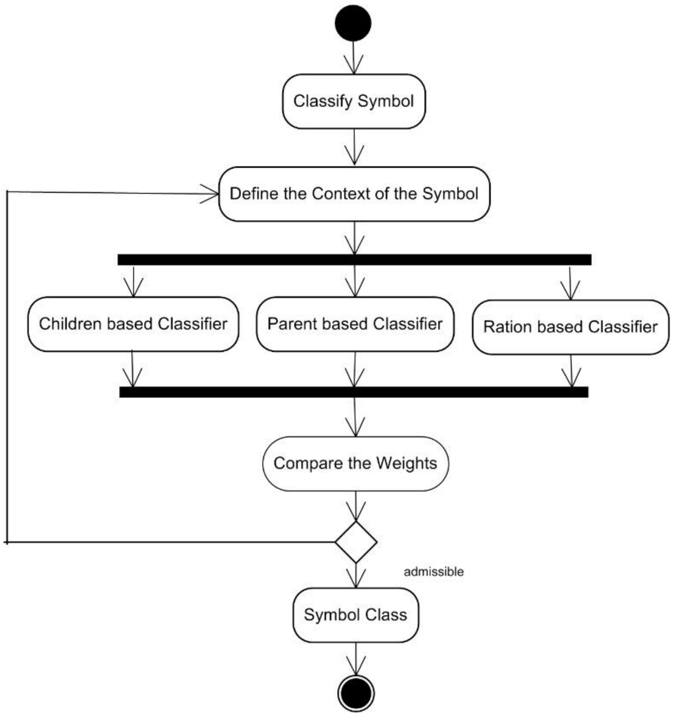


Fig. 2. Activity Diagram of Classifier of Symbols

Classifier of symbols classifies characters in one of four classes: "small", "upper", "under", "variable range". Only the bounding rectangles are considered and the value of the probability of the character of the symbol for each class is returned. For classification of characters, their context is considered (see Fig. 2).

The system being developed is a multiclassifier that adapts to each character, taking into account its context. Each classifier returns a set of probability values.

Classifier based on child elements. The classifier is made up of five neural networks, one for each child element. Four entries for each classifier: the child element class; relative vertical position; relative horizontal position; size relative to the parent element.

The classifier based on parenting elements looks at the position and size of the symbol relative to the parent. It also takes into account the parent element class and type of connection (e.g., index). The classifier is also a neural network.

A *ratio classifier* allows classifying a character regardless of its context, using only information about the restriction rectangle. For a rough character classification, the Bayesian system is used.

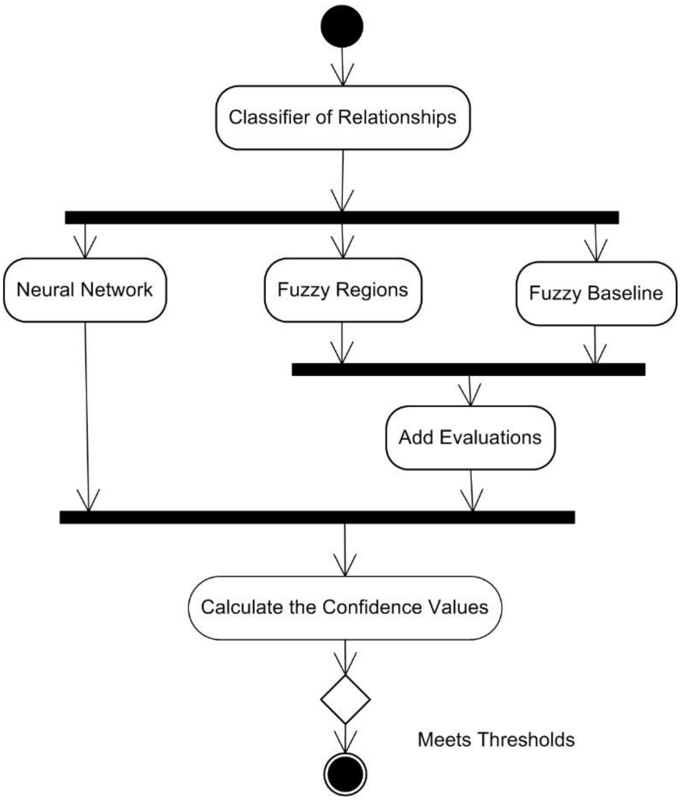


Fig. 3. Activity Diagram of Classification of Relationships

Classifier of relationships determines which is the most probable link between two characters. The probability value for each class is returned. The components of the classifier are the neural network, fuzzy areas and fuzzy baselines. The classifier consists of three independent parts, each of which gives the value of the probability of communication (see Fig. 3). The results are combined to give a final answer, which can then be compared with the limits.

The central part is the neural network (*Neural Network*). It is trained in data mining to effectively identify the relationships between two given sim-wolves. Input:

- H - relative size of the child element relative to the parent: $H = \frac{h_p}{h_c}$;
- D - relative vertical position of the child element: $D = \frac{y_p - y_c}{h_p}$;
- V - relative horizontal position of the child element: $V = \frac{xmax_p - xmin_c}{w_p}$, where the *p* and *c* indices are "parent" and "child", *h* is height, *w* is width, *y* is the vertical center of the restrictive rectangle, and *xmin* and *xmax* are the left and right boundaries.

The purpose of the *Fuzzy Regions* system is to help the neural network classify the relationship, and also indicate when the characters have no links. Fuzzy areas are used for all interconnections, except "embedded", where fuzzy baselines are used instead. To evaluate the confidence that a child character is in a certain relation to a possible parent symbol, we compute the membership value for the center of the left border of the child element in the corresponding fuzzy field of the parent element.

Fuzzy Baseline. Unlike other child elements, the built-in is not necessarily close to its parent's character. This makes the use of regions more difficult. The built-in child element is on the same line as its parent's character. The position of the base line of the character mainly depends on its class.

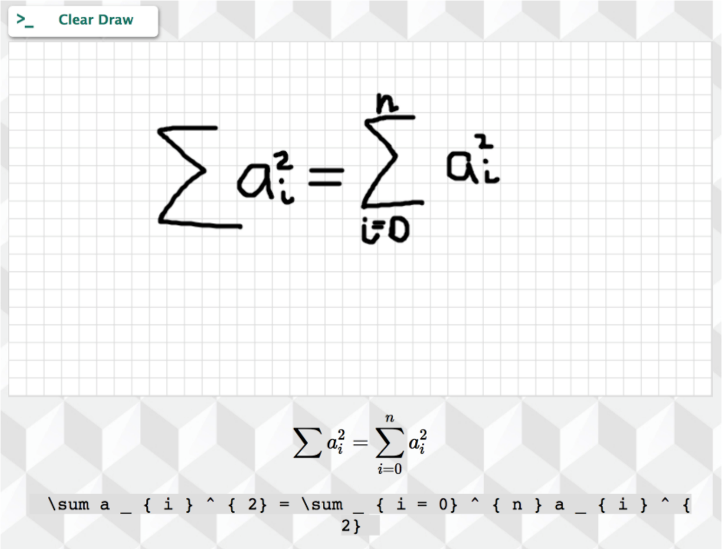
Since a flexible solution is been developed, the possible change of the line of writing by considering fuzzy baselines is processed. For a couple of parent / child symbols first the basic level of the parent element is computed, taking into account its class. Then, the possible baseline lines of the child character are considered, and the probability estimation is calculated based on the distance from the parent-baseline and the probability of the child-class.

5 Ingredients of the Intelligent System for the Recognition of Mathematical Expressions

Various paradigms and technologies for working with information resources have been analyzed for the project implementation [17-24]. An object-oriented approach to designing an information system is chosen. The intelligent information system can be divided into the following structural components:

- the main part, representing the expression and performing the classification,
- part of the input / output to avoid repeating the same things;
- graphical user interface.

The prototype of the experimental implementation of the intellectual system uses the frameworks (Weka and JLatexMath) and the database, which is implemented using MS Access.



$$\sum a_i^2 = \sum_{i=0}^{n} a_i^2$$

```
\sum a _ { i } ^ { 2} = \sum _ { i = 0} ^ { n } a _ { i } ^ {
2}
```

Fig. 4. Graphical user interface

Using a graphical interface makes use of the system and visualization of the results easier and more intuitive (see Fig. 4). Classes implementing the graphical interface built using Web technologies.

Testing of the system on a complete set of tasks was carried out. To ensure the speed of recognition, and also given that the average number of symbols of the mathematical expression is about 6, the variant of the algorithm with the optimal number of iterations is selected – 7. Successfully classified 441 of 570 characters (percentage of correct results – 77.36%).

The results of the analysis showed that simple structure recognition can help classify symbols. Characters can be properly classified in the presence of a sufficient context. In the case of an incorrect classification of the character, it was determined that the probability value for the correct class was also high enough. Recognition of the structure was fast and meets the requirements. A general analysis of the performed tests confirmed that the methods of machine learning allow recognizing the structure by comparing the characters by two.

6 Conclusions

The paper analyzes existing methods and approaches to the recognition of mathematical expressions. The possibility of simultaneous execution of structural analysis and classification of characters, using little knowledge about the syntactic system of the mathematical expression is investigated. The proposed approach is based on a mutual limitation between symbols and structure. Knowledge of the symbol value helps to analyze the structure, but the structure can help eliminate ambiguity in recognizing a character.

Instead of defining the characters, they are classified. Classification can be accomplished with the use of bounding rectangles of symbols and only the structure of the expression. An iterative algorithm is developed for the use of reciprocal constraints between the structure and type of characters. Although the classification of characters consists in the classification of each symbol separately, the recognition of the structure is a more complex task. Links between symbols must be found and identified. One-pass algorithm is implemented, which contains a search with return ion, which provided a quick recognition of the structure.

Expression recognition returns the probability value for each character and link, rather than a clear interpretation of the expression. Presenting results using probability values makes it easy to use the system as part of the larger one that performs all recognition. These probabilities are also used to determine estimates that give an idea of how good the system is. A flexible, adaptive method is implemented that returns the value of probability, and not a clear answer. A combination of neural networks is used to classify links between two symbols and estimates based on fuzzy baselines and fuzzy areas around the symbol.

The project of the intellectual system implements an iterative algorithm based on the methods of machine learning. A graphical user interface is created that allows using the expression recognition system quickly and easily.

7 References

1. Shapiro, L., Stockman, G.: Computer vision. Washington University (2006)
2. Veres, O., Rusyn, B., Sachenko, A., Rishnyak, I.: Choosing the method of finding similar images in the reverse search system. In: CEUR Workshop Proceedings. vol. 2136, Proc. of the Int. Conf. COLINS, vol. 1, pp. 99–107 (2018)
3. Rusyn, B., Lutsyk, O., Lysak, O., Lukeniuk, A., Pohreliuk, L.: Lossless Image Compression in the Remote Sensing Applications. In: Int. Conf. on Data Stream Mining & Processing (DSMP), 195-198 (2016)
4. Rashkevych, Y., Peleshko, D., Vynokurova, O., Izonin, I., Lotoshynska, N.: Single-frame image super-resolution based on singular square matrix operator. In: IEEE 1th Ukraine Conference on Electrical and Computer Engineering (UKRCON), 944-948 (2017)
5. Lytvyn, V., Vysotska, V., Veres, O., Rishnyak, I., Rishnyak, H.: Classification methods of text documents using ontology based approach. In: Advances in Intelligent Systems and Computing, pp. 229-240 (2017)
6. Zanibbi, R., Blostein, D., Cordy, J.: Recognizing mathematical expressions using tree transformation. In: IEEE Transactions on Pattern Analysis and Machine Intelligence 24(11), pp. 1455–1467 (2002)
7. Zhang, L., Blostein, D., Zanibbi, R.: Using fuzzy logic to analyze superscript and subscript relations in handwritten mathematical expressions. In: Eighth International Conference on Document Analysis and Recognition (ICDAR'05), vol. 8, pp. 972–976. (2005)

8. Tapia, E., Rojas, R.: Recognition of on-line handwritten mathematical expressions using a minimum spanning tree construction and symbol dominance. In: Graphics Recognition Algorithms and Applications (Lecture Notes in Computer Science). pp. 329–340 (2004)
9. Eto, Y., Suzuki, M.: Mathematical formula recognition using virtual link network. In: Proc. Sixth Int'l Conf. Document Analysis and Recognition (ICDAR 2001), pp. 762-767 (2001)
10. Rhee, T., Kim, J.: Efficient search strategy in structural analysis for handwritten mathematical expression recognition. Pattern Recognition 42(12)(12), pp. 3192–3201 (2009)
11. Miller, E., Viola, P.: Ambiguity and constraint in mathematical expression recognition. In: Proc. 15th National Conf. on Artificial Intelligence (AAAI 98), pp. 784-791 (1998)
12. Chen, Y., Shimizu, T., Okada, M.: Fundamental study on structural understanding of mathematical expressions. Systems, Man, and Cybernetics 2, pp. 910–914 (1999)
13. Awal, A., Mouchere, H., Viard-Gaudin, C.: Towards handwritten mathematical expression recognition. In: 10th International Conference on Document Analysis and Recognition, (ICDAR 2009), pp.1046-1050 (2009)
14. Wang, Z., Faure, C.: Automatic perception of the structure of handwritten mathematical expressions. Computer Processing of Handwritting, pp 337–361 (1990)
15. Aly, W., Uchida, S., Suzuki, M.: Identifying subscripts and superscripts in mathematical documents. Mathematics in Computer Science 2(2), pp. 195-209 (2008)
16. Aly, W., Uchida, S., Fujiyoshi, A., Suzuki, M.: Statistical classification of spatial relationships among mathematical symbols. In: 10th International Conference on Document Analysis and Recognition, pp. 1350-1354 (2009)
17. Lytvyn, V., Vysotska, V., Veres, O., Rishnyak, I., Rishnyak, H.: The Risk Management Modelling in Multi Project Environment. In: Computer Science and Information Technologies (CSIT2017), pp. 32-35 (2017)
18. Shakhovska, N., Bolubash, Yu., Veres, O.: Big Data Federated Repository Model. In: The Experience of Designing and Application of CAD Systems in Microelectronics (CADMS'2015), pp. 382-384 (2015)
19. Shakhovska, N., Veres O., Bolubash, Y., Bychkovska-Lipinska, L.: Data space architecture for Big Data managering. In: Computer Science and Information Technologies (CSIT2015), pp. 184-187 (2015)
20. Veres, O., Shakhovska, N.: Elements of the formal model big date. In: Perspective Technologies and Methods in MEMS Design (MEMSTECH'2015), pp. 81-83 (2015)
21. Lytvyn, V., Vysotska, V., Dosyn, D., Burov, Y.: Method for ontology content and structure optimization, provided by a weighted conceptual graph, Webology, 15(2), pp. 66-85 (2018)
22. Lytvyn, V., Peleshchak, I., Vysotska, V., Peleshchak, R.: Satellite spectral information recognition based on the synthesis of modified dynamic neural networks and holographic data processing techniques, 2018 IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 330-334 (2018)
23. Chen, J., Dosyn, D., Lytvyn, V., Sachenko, A.: Smart Data Integration by Goal Driven Ontology Learning. In: Advances in Big Data. Advances in Intelligent Systems and Computing. – Springer International Publishing AG 2017. P. 283-292 (2017)
24. Su, J., Vysotska, V., Sachenko, A., Lytvyn, V., Burov, Y.: Information resources processing using linguistic analysis of textual content. In: Intelligent Data Acquisition and Advanced Computing Systems Technology and Applications, Romania, 573-578, (2017)