UDC 004.855.5

OVERVIEW OF OCR TOOLS FOR THE TASK OF RECOGNIZING TABLES AND GRAPHS IN DOCUMENTS

O. Yaroshenko

This study describes OCR tools for recognizing tables and graphs. There is a great demand for solutions that can effectively automate the processing of an extensive array of documents.

Existing OCR solutions can efficiently recognize text, but recognizing graphical elements, such as charts and tables, is still in the making. Solutions that can increase the accuracy of visual data recognition can be valuable for technical document processing, such as scientific, financial, and analytical documents.

Key words: *OCR, PDF files, FastText, detection, recognition, deep learning, technical documents.*

Introduction

In the modern world, every day, a huge number of different documents are translated from paper to electronic form: printed texts, payment orders, customs or tax declarations, ballots, various questionnaires, and many others. Thousands of different electronic document management systems are actively used in almost all spheres of activity.

Thanks to general computerization and the spread of electronic document circulation in various areas of human activity, huge archives of textual and visual information have been accumulated. The global Internet is a continuously expanding electronic archive.

The analysis of modern information systems made it possible to draw a conclusion about the limited possibilities of semantic analysis and image search. Semantic analysis of images means automatically obtaining their semantic descriptions (annotations) and searching in the space of these descriptions (search by content) [4].

Among the search types implemented by information systems, image search by keywords is the closest to meaningful search but has one significant drawback - keywords for images are created by an expert. In all systems of electronic document circulation and systems of entering printed texts, one of the key stages is the recognition of text symbols - the translation of information from graphic form - the result of scanning - into text form. In most cases, the raw document data has noise in it, i.e., unwanted features that make the image hard to perceive. Although these images can be used directly for feature extraction, the accuracy of the algorithm would suffer greatly. This is why image processing is applied to the image to get better accuracy.

Despite the long history of the development of recognition algorithms and the existence of a large number of algorithms, clearly printed texts are recognized well. The problem of recognition in more complex cases is far from being solved [8].

There is a question of further increasing the accuracy of recognizing documents of poor quality; in particular, existing algorithms provide a relatively low accuracy of recognizing texts from graphic images obtained by scanning with small resolutions [1].

It is worth noting the class of problems in which graphic the image cannot be improved by increasing the scanning resolution or changing the scanning parameters. For this, papers (receipts, business cards, reports, internal decrees) are usually processed by a scanner, and OCR software creates searchable PDF files for the required text fragment.

Text recognition systems or OCR systems (Optical Character Recognition) are designed to automatically enter documents into a computer. It can be a page of a book, a magazine, a dictionary, or some kind of document - anything that has already been printed and needs to be converted back to electronic form [3].

Thus, the development of new high-precision text recognition algorithms, as well as the improvement of existing ones, is a potentially useful task.

OCR systems development

The history of the most massive demand for OCR systems began with the "competition" between CuneiForm and FineReader systems of the same version 1.3. According to many independent specialists, CuneiForm was more robust regarding the sum of indicators than [3].

The backbone of the development team of this program was based in the USA. However, unfortunately, even before the release of the CuneiForm 2.0 version, this team practically ceased to exist. Moreover, BIT kept its team of programmers [9]. OCR is used for two main tasks: document archiving and document editing. For this, papers (receipts, business cards, reports, internal decrees) are usually processed by a scanner, and OCR software creates searchable PDF files for the required text fragment [5].

Such programs usually convert a printed table into an Excel file or a paper document into an electronic one that can be edited and used later on a PC. Powerful OCR software can also convert printed text into HTML files. They can immediately be placed on the site for public access.

These tasks include previously created electronic archives of documents in the form of bitmap images, electronic libraries, and text messages. OCR is used for two main tasks: document archiving and editing [6]. When choosing an OCR program, one needs to decide whether it wants it to run automatically, interactively, or in combination with others. With the offline operation, the utility starts working immediately after scanning the document. A few seconds after processing the paper medium, the program produces the final result [2].

Of course, editors built into recognition systems cannot compete with Microsoft Word or Lotus Word Pro. OCR editors - programs are designed in such a way as to simplify the process of eliminating recognition errors and errors as much as possible: the system allows one to observe the "original" graphic image of the document during the editing process [2].

Almost all recognition programs have a built-in spell check system, even at the recognition stage. In the editor, "doubtful" symbols and words not in the dictionary are highlighted in a unique color.

When the document is edited, it can be saved as a file (TXT, RTF). The RTF format (Rich Text Format) is understandable by most word processors (Microsoft Word, Lotus Word Pro, Word Perfect, Lexikon). It allows one to specify information about the design (fonts, paragraphs, illustrations, columns, trimming, tables) [3].

The finished document can be transferred to the editor using the Drag&Drop mechanism or via the Clipboard. If the document contains tables, they can be written as Word tables or directly transferred to Excel spreadsheets.

OCR systems recognize text and various elements (pictures, tables) from an electronic image. The image is usually obtained by scanning a document and, less often, by photographing it. The algorithm of the OCR program processes the received image, areas of text, images, and tables are highlighted, and garbage is separated from the necessary data (Fig 1).

At the next stage, each character is compared with a unique dictionary of characters; if a match is found, this character is considered recognized (Fig. 2). As a result, one gets a set of recognized characters, that is, the desired text. Modern OCR systems are pretty complex software solutions [7].

After all, the text can be littered, distorted, or polluted, and the program must take this into account and handle such situations properly. In addition, modern OCR systems also make it possible to obtain a copy of a printed document in electronic form, preserving formatting, styles, text sizes, and fonts.

Description of the OCR procedure

1. Image pre-processing.
2. Recognition of objects of higher levels.
3. Character recognition
4. Hypothesis structuring. Vocabulary check.
5. Synthesis of an electronic document.

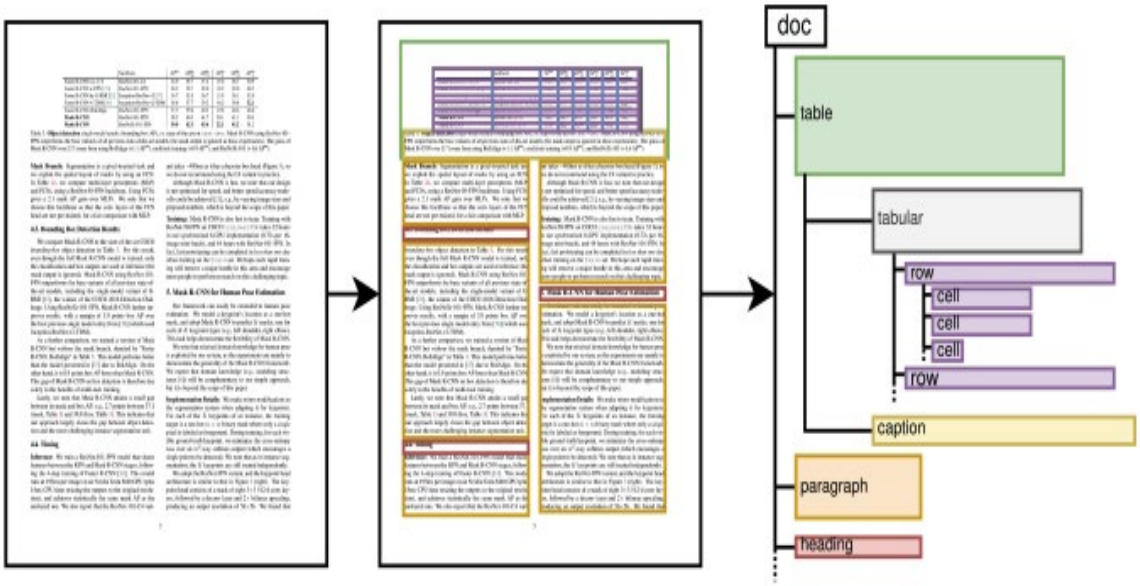

Fig. 1. Document structure recognition [18]

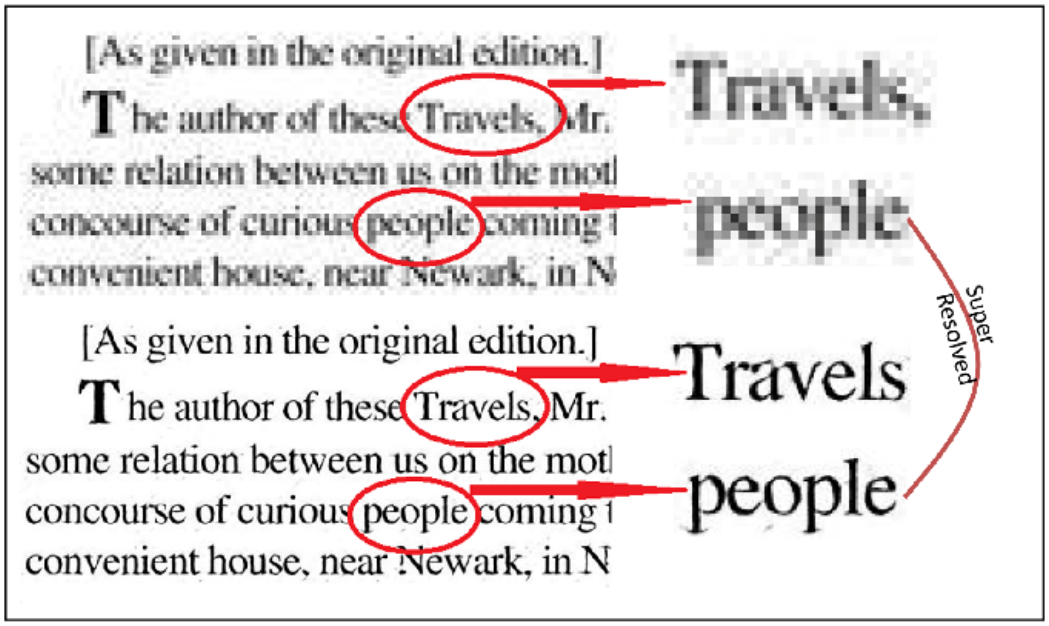

Fig. 2. Character identification and recognition example [13]

Most OCR Optical Character Recognition programs work with a bitmap image received through a fax modem, scanner, digital camera, or another device. The first step in OCR is to break up the page into blocks of text based on the particularities of right and left alignment and the presence of multiple columns. The recognized block is then split into lines [7].

As a result, there is a problem determining the line to which this or that image fragment belongs. For example, for the letters j, with a slight slope, it is already difficult to determine which line the upper (separate) part of the character belongs to (in some cases, it can be mistaken for a comma or a period). The lines are then broken up into contiguous regions of the image, which generally correspond to individual letters; the recognition algorithm makes assumptions about the correspondence of these regions to characters; and then a selection of each character is made, as a result of which the page is restored in characters of text, and, as a rule, in the appropriate format. OCR

systems can achieve the best recognition accuracy of over 99.9% for pure images composed of regular fonts [4].

At first glance, this recognition accuracy seems ideal. However, the error rate is still depressing because if there are approximately 1500 characters per page, then even with a recognition success rate of 99.9%, there are one or two errors per page. In such cases, the dictionary check method comes to the rescue.

If a word is not in the system's dictionary, it tries to find a similar one according to special rules. However, it still does not allow to correct 100% of errors, which requires human control of the results.

The modern state of OCR processing for technical documents

Heavy use of PDF files has promoted research in analyzing the file layout for text extraction purposes. One of the PDF document's difficulties is that smartphone users extensively scan the documents in PDF format using the phone camera. Optical Character Recognition (OCR) techniques must be employed to get these images into text format [11]. OCR is a technology still in the making, and available software provides varying levels of accuracy. The best results are usually obtained with a tailored solution involving corpus-specific pre-processing, model training, or postprocessing, but such procedures can be labor-intensive. Pre-trained, general OCR processors have a much higher potential for wide adoption in the scholarly community; hence, their out-of-the-box performance is of scientific interest. Modern OCR framework comparison research indicated that certain types of "integrated" noise, such as blur and salt and pepper, generate more errors than "superimposed" noise, such as watermarks, scribbles, and even ink stains (Fig 3).

Furthermore, it suggests that the "OCR language gap" persists. Calls for special efforts to improve the quality of document images before passing them to the OCR engine. [12] One compelling option is to super-resolve these low-resolution document images before passing them to the OCR engine. Experiments show an improvement of up to 21% in accuracy OCR on test images scanned at low resolution. One immediate application of this can be enhancing the recognition of historical documents scanned at low resolutions [13].

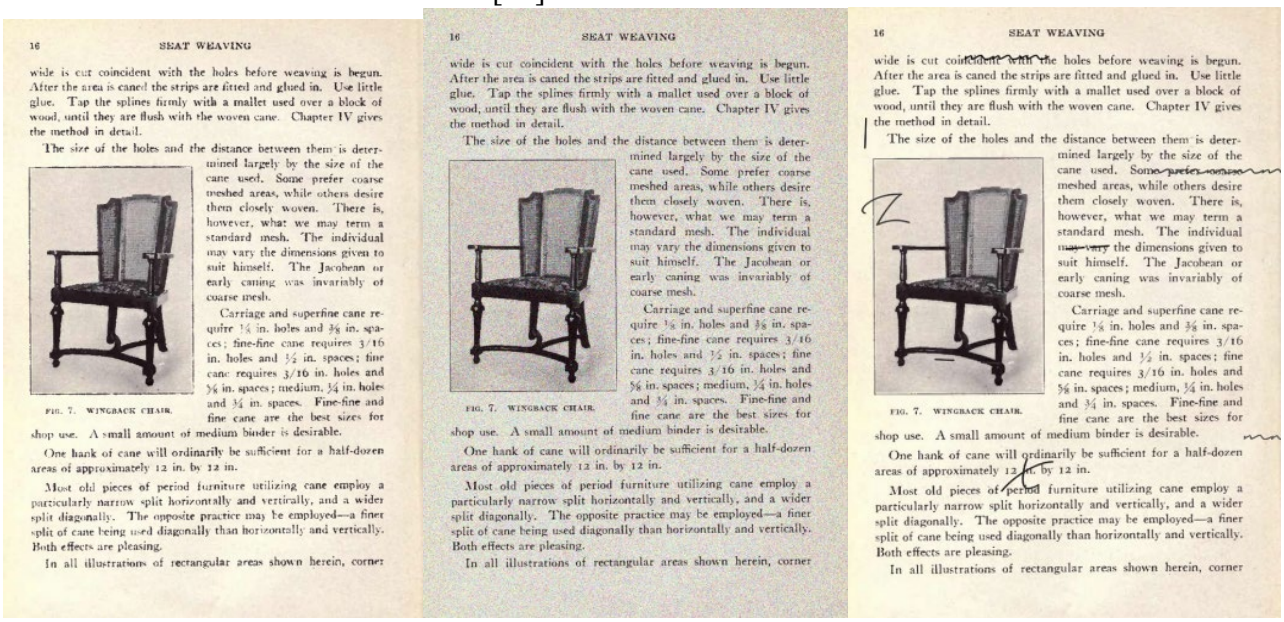

Fig. 3. Document noise examples

Scientific papers and other technical documents are composed of natural language text and other modalities, like block diagrams, mathematical formulas, tables, graphics, and pictures. Automatic Technical Documents Processing and Understanding (TDPU) has received more attention in the last two decades due to its profound applicability. TDPU represents the continuation of the progress made in the fields of OCR, Natural Language Understanding, Pattern Recognition, and Image Understanding. [14]

Research activities in document image analysis can be mainly classified into two categories, text processing, and non-text processing, e.g., figures, graphics, and diagrams [19]. Although the introduction of optical character recognition technologies mostly solved the task of converting human-readable characters from images into machine-readable characters, the task of extracting table semantics has been less focused on over the years [16]. Also, chart recognition techniques for document images are still an unsolved problem due to the great subjectiveness and variety of chart styles [19].

The recognition of tables consists of two main tasks, namely table detection and table structure recognition (Fig 4). There are works that recognize table structures from text or other syntactic tokens rather than directly from document renderings. One draws upon deep neural networks to identify table structures for rendered inputs. The proposed architecture combined the benefits of convolutional neural networks for visual feature extraction and graph networks for dealing with the problem structure. They empirically demonstrated that their method outperforms the baseline by a significant margin. However, they aim at a different purpose: parsing table structures but not complete document hierarchies. As such, the authors do not attempt to identify text elements or nested figures. [17].

Research regarding mathematical formula detection identified that the key difference between formula detection in typeset documents and object detection in natural scenes is that typeset documents avoid the occlusion of content by design. This constraint may help design a better algorithm for non-maximal suppression, as the original non-maximal suppression algorithm is designed to handle overlapping objects. They believe improved pooling will reduce the number of over-merged and split detections, improving precision and recall. This approach can detect not only formulas but also other types of structures in technical documents [10].

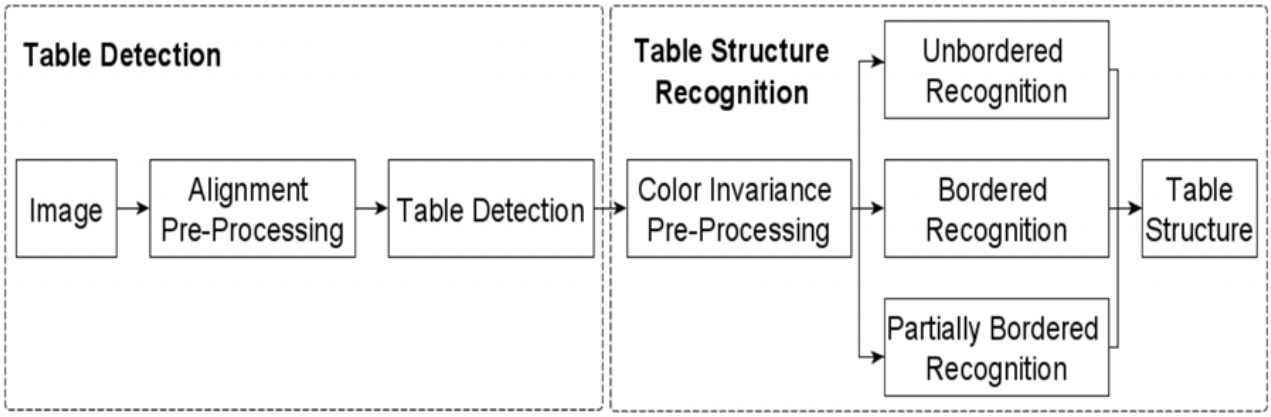

Fig. 4. The two-stage process of TD and TSR in Multi-Type-TD-TSR. [16]

Prasad et al. (2020) developed a model for table structure detection based on CNN architecture which was originally trained for objects in natural scene images and was also very effective for detecting tables. Moreover, iterative transfer learning and image augmentation techniques can be used to learn efficiently from a small amount of data. The proposed model recognized structures within tables by predicting table cell masks while using the line information. It was stated that improving the post-processing modules can further enhance the accuracy [15].

For this purpose, Fisher et al. (2021) distinguished three types of tables (Fig 5), depending on whether they are borderless or not. Because of the unavailability of large labeled datasets for table structure recognition, they decided to use two conventional algorithms: The first one that can handle tables without borders and the second one that can handle tables with borders. Further, they combined both algorithms into a third conventional table structure recognition algorithm that can handle all three types of tables. This algorithm achieves the highest F1 score among the systems compared in their research for an IoU threshold of 0.6 and 0.7 but does not detect sharp borders, so the F1-score decreases rapidly for higher thresholds of 0.8 and 0.9 [16].

Rang	Team
1	Centurion
2	Pinbu$taz
3	Kugelblitz
4	Cosinus phi
5	Rattlesnake on Tour
6	Dark Pins
7	Strike Sharkattack
8	Holy Wings
9	Alfi und die Chipmunk

a) b) c)

Fig. 5. Types of tables based on how they utilize borders: a) tables without borders, b) tables with partial borders, c) tables with borders [16]

Rausch et al. (2021) presented a solution that takes rendered document images as input, performs segmentation into bounding boxes, and then outputs the hierarchical structure of the entire document. Their solution identified table and tabular elements with high precision, but other elements were recognized with significantly lower accuracy. They emphasize again that both suitable baselines and datasets for this task are hitherto lacking [18]. Qasim et al. (2019) also identified the lack of large-scale datasets as a significant hindrance to deep learning research for structure analysis. They presented a new large-scale synthetic dataset for the problem of table recognition [17]. Ayinala and Grandhi (2021) stated that text processing is an essential task as we have more digital content available on the Internet today. The most challenging task nowadays is locating and analyzing textual information [11].

Another big problem with the correct processing of technical documents is chart recognition. They are widely used to represent numeric and qualitative data in different formats. Although existing OCR tools can recognize the text content of chart segments, the primary data represented by the chart, which is usually shown visually by lines, bars, and circle segments, is not recognized well by those tools. The main issue is that there are plenty of different chart types and styles for each particular chart type, and most research is focused on a limited set of charts representation [19].

For the task of extracting data from chart images, the detection process is a preliminary step. It helps to locate and extract the data chart only and classify chart type, improving data recognition performance. For such tasks, Convolutional Neural Networks (CNN) are commonly used. CNN-based methods show outstanding results in various object detection domains. There is a lack of works in the literature linking real-world photos with the task of labeling charts before labeling. There are many issues to solve, such as locating charts in images and removing camera distortions [20].

It can be said that more advanced solutions for chart recognition are a necessary addition to existing OCR systems [19].

Conclusion

OCR systems recognize text and various elements (pictures, tables) from an electronic image. The image is usually obtained by scanning a document and, less often, by photographing it. The algorithm of the OCR program processes the received image, areas of text, images, and tables are highlighted, and garbage is separated from the necessary data. At the next stage, each character is compared with a unique dictionary of characters; if a match is found, this character is considered recognized. As a result, one can get a set of recognized characters, that is, the desired text.

As described above, technical document processing is a demanding and underdeveloped area of deep learning. There have been many types of research in this area in recent years, but the main focus is on the text and common pattern recognitions inside documents. On the other hand, technical

documents have a lot of specific structures (mainly charts and tables) inside and require a high recognition accuracy to be considered.

Directions for future research

There are several gaps in the technical document processing research that follow from the findings in this article that would benefit from further research, including extending and further testing statements developed here:

1) In-depth exploration of how OCR algorithms can be re-evaluated and modified to perceive scanned PDFs with technical documentation better. That may include new approaches to removing noise from scanned images and solutions for document content structure recognition.

2) Gathering new datasets of scanned PDFs that include specific elements, such as tables and charts of different types, to enhance further models that work with processing such elements.

3) Improing unique structure recognition and processing methods by comparing existing deep neural networks with different architectures for such tasks. Based on the results, it would be beneficial to build data pipelines that will combine different methods to improve the final solution's accuracy.

References

1. Gorelik A. L. Recognition methods / A. L. Gorelik, V. A. Skripkin. – M. : High School, 1984. – 219 p.
2. Voloshyn G. Ya. Pattern recognition methods. Book 2 / G. Ya. Voloshyn, A. A. Ilyin.
3. Pisarevskiy A. N. Systems of technical vision (fundamental principles, hardware and software) / A. N. Pisarevskiy, A. F. Chernyavskiy, G. K. Afanas'ev. – Leningrad : Mechanical engineering. Leningrad department, 1988. – 423 p.
4. Babak V. P. Obrobka signals: Handyman / V. P. Babak, V. S. Khandetsky, E. Schryufer. – Kiev : Libid, 1996. – 392 p.
5. Engineering drawing / edited by cand. tech. science prof. G. P. Vyatkina. – Mechanical engineering, 1985. – 368 p.
6. Blatner D. Scanning and rasterization of images / D. Blatner, G. Fleishman, S. Rot. Per. from English. – M. : EKOM Publishing House, 1999. – 400 p.
7. Forsyth D. A. Computer Vision: A Modern Approach. 2nd Edition / D. A. Forsyth, J. Pons. – Williams Publishing Center, 2004. – 928 p.
8. Mikheeva E. V. Information technology in professional activities / E. V. Mikheeva. – M. : Academy, 2007. – 384 p.
9. Grebenyuk E. I. Technical means of informatization: Textbook for environments. Prof. education / E. I. Grebenyuk, N. A. Grebenyuk. – M .: Publishing Center «Academy», 2005. – 272 p.
10. Mali P. ScanSSD: Scanning Single Shot Detector for Mathematical Formulas in PDF Document Images [Electronic resource] / P. Mali, P. Kukkadapu, M. Mahdavi, R. Zanibbi – Access Mode: https://arxiv.org/abs/2003.08005
11. Ayinala H. K. Text classification from PDF documents / H. K. Ayinala, S. Grandhi // In International Research Journal of Modernization in Engineering Technology and Science, 2021. – Vol. 3. – 58-63 pp.
12. Hegghammer T. OCR with Tesseract, Amazon Textract, and Google Document AI: a benchmarking experiment / T. Hegghammer // Journal of Computational Social Science, 2022. – Vol. 5. – 861-882 pp.
13. Lat A. Enhancing ocr accuracy with super resolution / A. Lat, C. V. Jawahar // In 2018 24th International Conference on Pattern Recognition (ICPR) (20-24 August 2018). – IEEE, 2018. – 3162-3167 pp.
14. Kostalia E. E. Evaluating Methods for the Parsing and Understanding of Mathematical Formulas in Technical Documents / E. E. Kostalia, E. G. Petrakis, N. Bourbakis // 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI) (09-11 November 2020). – IEEE, 2020. – 407-412 pp.

15. Prasad D. CascadeTabNet: An approach for end to end table detection and structure recognition from image-based documents / D. Prasad, A. Gadpal, K. Kapadni, M. Visave, K. Sultanpure // In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (June 2020). – IEEE, 2020. – 572-573 pp.
16. Fischer P. Multi-Type-TD-TSR – Extracting Tables from Document Images using a Multi-stage Pipeline for Table Detection and Table Structure Recognition: from OCR to Structured Table Representations / P. Fischer, A. Smajic, A. Mehler, G. Abrami // Lecture Notes in Computer Science, 2021. – Vol. 12873.
17. Qasim S. R. Rethinking table recognition using graph neural networks / S. R. Qasim, H. Mahmood, F. Shafait // In 2019 International Conference on Document Analysis and Recognition (ICDAR) (Sept. 20, 2019 to Sept. 25, 2019). – IEEE, 2019. – 142-147 pp.
18. Rausch J. DocParser: Hierarchical Structure Parsing of Document Renderings / J. Rausch, O. Martinez, F. Bissig, C. Zhang, S. Feuerriegel // In 35th AAAI Conference on Artificial Intelligence (AAAI-21) (May 2021). – AAAI-21, 2021. – 4328-4338 pp.
19. Liu Y. Review of chart recognition in document images. Visualization and Data Analysis / Y. Liu, X. Lu, Y. Qin, Z. Tang, J. Xu // Proceedings of SPIE – The International Society for Optical Engineering, 2013. – Vol. 8654. – 384-391 pp.
20. Araújo T. A Real-World Approach on the Problem of Chart Recognition Using Classification, Detection and Perspective Correction / T. Araújo, P. Chagas, J. Alves, C. Santos, B. Sousa Santos, B. Serique Meiguins // Sensors (Basel). – Vol. 20 (16). – 4370 p.