Computer Science Journal of Moldova, vol.**28**, no.3(84), 2020

# Semi-automated workflow for recognition of printed documents with heterogeneous content

Alexandru Colesnicov, Ludmila Malahov,
Svetlana Cojocaru, Lyudmila Burtseva

**Abstract**

The paper discusses problems of heterogeneous texts digitization. The archives of scanned printed documents grow dramatically by results of projects concerning cultural heritage preserving. Manual annotations of scanned document images and per page screen reading make the usage of these archives difficult and, sometimes, impossible. Existing document processing systems cannot automatically display content correctly due to the presence of heterogeneous content. We proposed a Web platform to maximize the support of semi-automated work of all used tools for recognition of heterogeneous documents. Maximizing support means both creating the convenient "single window" access to all tools, and reducing the manual part of the process as much as possible. For implementation, the convergent technology is used, which assembles complex software systems from ready-made heterogeneous modules on a single platform.

**Keywords:** platform for heterogeneous document recognition, page layout analysis, non-textual content recognition

## 1 Introduction

In most documents, whether old or contemporary, there are, along with the text, other elements: mathematical and chemical formulas, musical scores, diagrams, schemes, etc. In the process of digitizing such documents it is necessary to have an instrument, which would provide support not only for the recognition of the text itself, but also of other

components of a heterogeneous nature. For example, see collections of the scanned newspapers that are really huge archives [1] with heterogeneous content. Encyclopedia is a good example of a document with heterogeneous content, because we meet on its pages not only text, but a variety of content types including images, mathematical and chemical formulas, musical scores, technical drawings, chess notation, electronic circuits, etc.

Let's define the notion of heterogeneous content by associating it with the possibility of presentation in a scripting language [2]. The main features of such content are the following:

- the document is not exclusively in natural language;
- there is one or more scripting languages for presenting its components;
- the graphic representation can be rendered by script.

The modern active development of digitization of standard textual documents scans results in obtaining many robust and efficient solutions of this problem. Naturally, the researches employ already developed methods in solving the more general problem of heterogeneous content digitization. Nevertheless, heterogeneous content causes very specific problems which cannot be solved just by adapting of pure textual scans processing methods. Speaking generally about applied methods, the majority of them uses Deep Learning with additions or improvements which apply other techniques: dynamic programming [3], SVM [4], etc. A review of 29 techniques of script text mining with their accuracy estimation is presented in [5].

A number of latest research works present full described frameworks [4], [6]–[9]. The architecture of the presented frameworks depends on functionality: whole archive digitization or partial processing by request. These two groups frameworks are constructed inside almost identically. Basing on this, we can in general consider the architecture and the workflow of such framework as having defined pattern. Online services can either be included in the general framework or they create own online framework [10], [11]. The set of services requested by researchers of focus domain can also be considered as defined.

Recognition of documents with heterogeneous content is quite com-

plex and requires complex analysis of the image with division into homogeneous segments. The research uses already developed methods to solve the general problem of recognizing homogeneous content. Heterogeneous content causes very specific problems that cannot be solved simply by adapting methods for handling pure text scans. Generally speaking on applied methods, most of them use Deep Learning with additions or improvements.

Therefore, there is a need to create a framework that would provide support in digitizing heterogeneous texts, automating processes where it is possible and interacting with the user when the manual intervention or an expert opinion is necessary.

## 2 Scripting languages

We investigate formal presentation of graphical content with the aim of automated optical recognition. We need a term with wide coverage that includes formal languages for presentation of graphical content. The term "scripting languages" is used narrowly referring to a subclass of programming languages. Another term is "markup languages" but it is restricted by several specific areas.

Heterogeneous content includes graphical elements, in addition to the pure text. As to the graphical content, we exclude from our investigations pure images presented by bitmaps. The object of our research is the graphical content that may be presented by a description (script) in a formal language without loss of information, and may be restored from the said script. To be shorter, we will use the term "scripting languages" instead of more precise "formal languages for presentation of graphical content" in the rest of our paper.

Scripting languages were developed mainly in specific application areas, like mathematics and physics, chemistry, music, building and architecture, technical design, chess, etc.

Many features of scripting languages are defined by specific application area. For example, an important issue for chemistry is the unequivocal identification of a chemical for production and trade. The assortment of chemicals is huge (millions). Each of them can have

dozens of trade names, and, for structurally complex chemicals, tens and hundreds image variants. See [2] for details.

An approach to universal scripting language is LaTeX. It was developed with orientation to mathematics but was extended to many areas including vector graphics, chemistry, music, chess, etc. due to its unrestricted extensibility. This language is constantly being developed and supplemented, in particular in terms of the presentation of non-textual elements. The output from a LaTeX compiler may be used in any device provided with the corresponding driver.

Many standardized scripting languages are extensions of the XML markup language, for example: MathML[1] for mathematics, and MusicXML[2] for music [12].

A standardized language used to exchange CAD projects in the electronic form is STEP (ISO 10303, Standard for the Exchange of Product model data). The covered areas are mechanical engineering, architecture, and building.

## 3 On digitization of heterogeneous content

### 3.1 Best practices

Heterogeneous content digitization problem consists of several subproblems, each of which is enough complicated to be a subject of autonomic research. Main groups of subproblems corresponding to processing steps are: pre-processing; recognition/layout analysis; assembling and saving to digital form. Only the first group of subproblems is well studied today.

The pre-processing of scans before content analysis is a usual step of digitization [13]. The main stages of pre-processing are the enhancement and binarization of document images. Document image enhancement here is the improving of the perceptual quality for maximum restoring of document initial look, the binarization is the separation of texts and background. Pre-processing has the specific aspect that can

[1] https://www.w3.org/Math/
[2] https://www.musicxml.com/

be simply formulated as: what is the noise in particular case. We have an experience of using the specialized tools for image preparation (for example, Scan Tailor). OCR programs have internal features for this.

Full range of image processing methods is used to make easier the following recognition stage. For example, work [14] applied the edge enhancement to increase visibility of content elements borders.

The problem of saving in digital form of textual historical documents was mostly resolved. The heterogeneous content digitization has to result in saving digital representation of layout and all its elements. Today digital representation exists for majority of type of content. Moreover, there are well described rules of such digital representation creating. Even very specific content can be supplied by digital representation without new developing. For example, a detailed study of digital representation of equations can be found in [15].

### 3.2 Document structure and layout analysis

The main problem of digitization is layout analysis. Document pages are divided into areas with the same type of content. The complex page physical layout is converted into logical structure. Layout analysis supposes region segmentation and region classification.

Region segmentation was thoroughly researched. Several segmentation algorithms were introduced being classified as top-down and bottom-up algorithms [16]. Later hybrid algorithms were proposed combining both approaches plus split-and-merge strategy [14], [17].

The problem of the identification and analysis of segment content for labeling heterogeneous components in different types of documents was investigated. Many variants of page structure analysis that provides possibilities to automate recognition of heterogeneous content were proposed. See [18]–[20].

Starting point of general text recognition is the defining of lines of text. For calligraphy of Arabic type all signs located between lines are significant [21].

For standard layout case, the digitization of heterogeneous documents applies text/non-text separation only for specific use case: news-

papers [6]. The newspaper has text and illustrations organized in strict, usually rectangular, shapes. In the contrary, historical documents often have overlapping elements [22].

The classification of elements in heterogeneous documents was proposed by [23].

Second specific feature of heterogeneous content digitization is recognition of pure non-textual elements. One can find studies related to specific elements: figures [10],[24],[25]; molecular structures [20],[26]; tables [27],[28], etc.

However, the general solution for the whole range of problems was not found and not implemented. It is necessary to develop a specific platform combining automated, semi-automated, and manual work.

### 3.3 Systems for recognizing homogeneous content

There are many OCR tools, commercial and open source, offline and online. OCR for text-based print content is the most advanced in terms of automatic recognition capabilities, user services, and validation of recognized texts. Examples are commercial ABBYY Finereader(AFR)[3], and open source free OCRopus [4] and Tesseract[5].

Currently, OCR tools are based on two technologies, character patterns and neural networks. Development of neural networks permitted to evolve text recognition from recognition of separate characters to the intelligent methods of recognition of words, lines, or phrases at once.

These OCR systems support restricted layout analysis of page images. For example, OCRopus has such features as: binary morphology; adaptive thresholding; deep learning based skew correction; page frame detection. Least square baseline finding algorithm is for constrained text line finding and modeling.

AFR provides the entire document recognition cycle, starting with scanning. It provides complex image correction, page segmentation with automatic detection of the segment type (restricted by text, ta-

[3]https:/www.abbyy.com/en-eu/finereader/tech-specs/
[4]https://github.com/tmbdev/ocropy/
[5]https://github.com//tesseract-ocr/

ble, or picture), analysis of the table structure, hyphenation detection, correction against dictionary, manual editing after recognition, etc.

OCRopus and Tesseract are packages of separate programs managed from the command line. Tesseract OCR may be used as a standalone program, or through API as a subsystem extracting text from images. Tesseract is compatible with many popular programming languages (C++, JavaScript, Python). Newer versions of Tesseract return not only recognized text but co-ordinates of recognized words on the page [6]. Tesseract.js[7] is a tool to extract text from images using JavaScript from HTML pages. Python-tesseract[8] is a Python wrapper for Google's Tesseract-OCR.

There are other text recognition systems but many of them are no longer supported. For example, CuneiForm was discontinued since 2008. Examples of supported OCR systems are AFR, Tesseract, OCRopus, CIB OCR[9], OCR.space (online)[10], Infty Reader[11], etc.

Recognition of heterogeneous content needs to use different software depending on the problem being solved. To recognize pure non-textual elements (figures, multi-chart, equation, diagram, photo, plot, table, art, and technical drawings), it is necessary to develop techniques, create tests collections, and to develop an integrated platform.

### 3.4 Practices selected as basis of implementation

Presented analysis produces the basis for our decisions about semi-automated workflow organization. For implementation, the technology of assembling complex software systems from ready-made heterogeneous modules on a single platform was tested; each of modules performs a small part of the task, and modules are combined using

[6] https://medium.com/jaafarbenabderrazak.info/ocr-with-tesseract-opencv-and-python-d2c4ec097866

[7] https://proglib.io/p/tesseract-js-izvlekaem-tekst-iz-kartinok-s-pomoshchyu-javascript-2020-04-22

[8] https://pypi.org/project/pytesseract/

[9] https://ocr.team/

[10] https://ocr.space/

[11] https://www.sciaccess.net/en/InftyReader/

Docker[12].

Deep learning is selected as the execution method firstly because of existing software solutions for many necessary subtasks. The analysis discovers the widespread practice to combine previously developed methods and techniques with standard deep learning for obtaining better solution, which is very suitable for our goals.

Architecture and the workflow of the platform we intend to develop can be inferred from analyzed works.

We selected Python as the language of implementation because it provides a lot of ready solutions in its rich libraries. Python with orientation to the Natural Language Processing is presented in [29], [30]. Python packages are used to develop modules for: preliminary preparation of scanned documents (cleaning, alignment, resolution, etc.); the analysis and scanning process; parsing requests to generate processing sequences; image analysis to obtain a page template + metadata using manual intervention to determine the type of context (images can contain formulas, graphs, notes, etc.); recognition of individual components and obtaining a pattern of the restored page; page assembling; manual correction; obtaining results with their XML description.

# 4 Semi-automated workflow for recognition of heterogeneous documents

## 4.1 Problems and challenges

From the above, we can draw the following conclusions:

- It is very difficult to recognize many kinds of heterogeneous content,
- Analysis of page structure is a complicated task,
- We need to integrate different kinds of scripts in a unified script presentation of the document.

Therefore, a need for a tool, supposedly, a Web platform to support semi-automated work over heterogeneous documents, arises. Despite a

[12] https://www.docker.com/

lot of achievements, automated recognition of the heterogeneous content remains a difficult problem. Our goal is to maximize the support of semi-automated work.

We see the process in execution of required actions: download input scan; read request; run cleaner; execute OCR including page layout analysis; execute request processing; publish answer.

To solve this problem, we propose a platform for recognition of heterogeneous documents, which uses the previously described and newly developed software, and can perform all stages of processing.

## 4.2 The functionality of a platform for recognition of heterogeneous documents

### 4.2.1 The design

We propose a design of a platform for recognition of heterogeneous documents to maximize the support of all processing steps. Maximizing support means both creating the convenient "single window" access to all tools, and reducing the manual part.

The recognition of heterogeneous documents involves many subtasks. Some of them may be performed automatically using specialized software. Other subtasks need slight manual intervention or manual control. If the specialized software does not exist, the processing is executed manually under the general purpose software.

Therefore, we can group all involved subtasks as follows:

**Automated:** scan; segment recognition according to types of segments; assembling of script presentation of pages with metadata integration; reconstruction of page images from scripts; automated verification.

**Semi-automated:** image quality improvement; page layout analysis; task distribution for manual verification.

**Manual:** expert verification and manual correction.

The platform is the Web framework with backend and frontend (Figure 1). Each of processes that constitute the platform functionality is executed by programs of different types. Corresponding to their

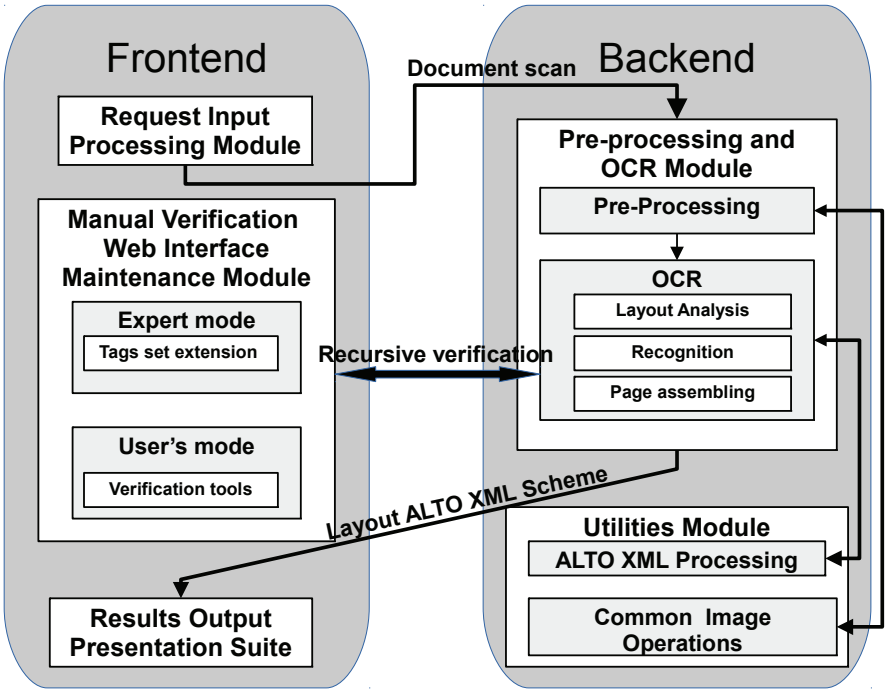



Figure 1. Main platform workflow

functionality, software is organized in container-based modules.The details of the platform functionality are described below.

### 4.2.2 Utility (backend)

The platform functionality supposes that modules are connected only by data transfer. For data representation, we chose ALTO (Analyzed Layout and Text Object) XML. The main valuable feature of ALTO XML is a good support of extensions, which would provide set of tags for new layout block types. Utilities include many commonly used tools like XML processing, image processing (ImageMagick), etc.

### 4.2.3 Input request (frontend)

This module aims to obtain user's input data. This may be only the document scan, but additional information like metadata can be supplied too. The module provides uploading and warehousing of the scans, and scheduling them for the next stage.

### 4.2.4 Pre-processing and OCR (backend)

Pre-processing of heterogeneous documents has specific features. Scans of such documents come mainly from Big Data archives, for example, created by projects in cultural heritage. All kinds of document images have the usual problems: noise, stains, shadings, distortions, etc. Pages can be damaged and can have different textures. During enhancement, we should not lose information.

Then the document scan is processed by a developed module described in details in 4.3. This script analyses the image and prepares page segment files and page maps. Page segments are recognized if possible. The next stage is the verification (automatic, semi-automatic, or manual). Finally, the assembling of the page scripts is performed. The result is a script presentation of pages of the document. This workflow is recursive as verification may unveil errors.

### 4.2.5 Manual verification (frontend)

Manual verification can be performed either by experts in the corresponding areas or directly by user depending on the particular requirements. Expert correction joins frontend and backend processes. It implies that the platform will be Web-based.

### 4.2.6 Results output (frontend)

This module executes a set of processes charged with the results presentation. At the previous stage, the recognized layout of the document scan was saved digitally as ALTO XML file. The module reconstructs source image and rebuilds layout scheme from ALTO XML

description. The result is generated to present the reconstructed document with availability of other details like ALTO XML page maps, metadata, annotations, etc.

### 4.3 Partitioning and mapping of heterogeneous documents

As discussed in 3.2, parsing a heterogeneous document is challenging. We tried some of the OCRs listed in 3.3. It turned out that the most complete tools are provided by ABBYY FineReader Engine (FRE). The set includes a ready-made command line interface (FRE CLI) utility with full recognition cycle. Unlike the standard AFR, the FRE CLI utility processes only one page at a time. The result is returned in XML format and contains the coordinates of the page segments, their type (text, image, table, separator), and, for text segments, also the recognized text. Thus, it is possible to cut the page into segments by type.

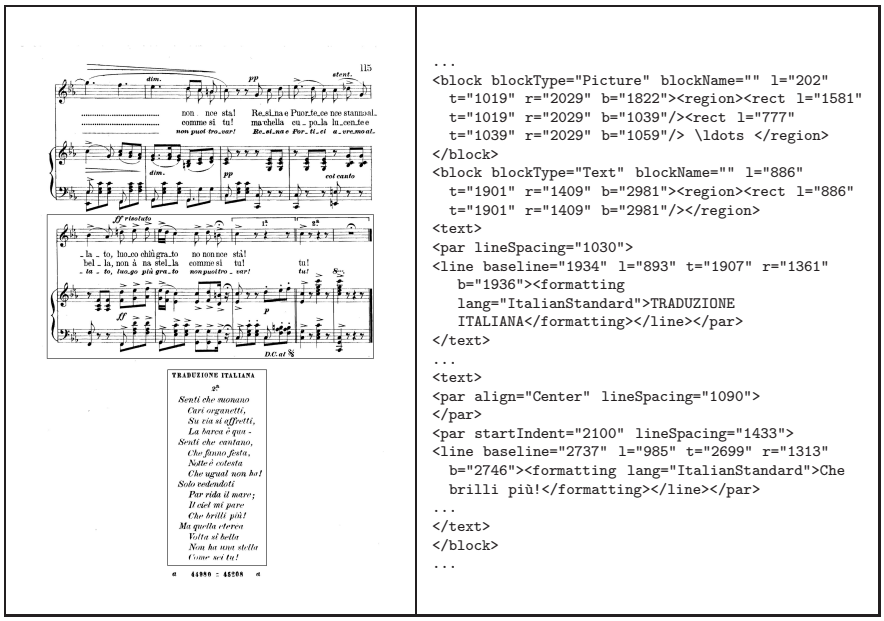
To process many pages, the utility can be called in a loop from a command script in any suitable language, including Python. FRE can be included in a Docker container.

A Python program was developed and coded to cut a scan of a document into segments with the same content. The process algorithm is as follows:

1. Using the FRE CLI utility over a page scan, we get an XML file with the coordinates of the upper and lower corners of the segment rectangles and the segment types (text, picture, table, etc.).
2. A program selects from the XML file the segment metadata (coordinates, segment types) and calls the image cutting utility.
3. A batch of scans of a multi-page document is processed in a cycle. File names with page images are set on the command line, with the ability to use regular expression placeholders “*” and “?”. For each image, a directory is created with a name derived from the image name, into which the XML file and page segments are placed in a format that matches the page image format. Sepa-

rators are excluded from processing. Operations over images are performed by the ImageMagick batch utility.

Thus, the program developed on the basis of FRE generates files with scans of document segments and XML files with page maps for further processing by the platform. Figure 2 on page 235 presents an example of a page scan and a fragment of an XML file after processing by the FRE CLI utility.




```
...
<block blockType="Picture" blockName="" l="202"
  t="1019" r="2029" b="1822"><region><rect l="1581"
  t="1019" r="2029" b="1039"/><rect l="777"
  t="1039" r="2029" b="1059"/> \ldots </region>
</block>
<block blockType="Text" blockName="" l="886"
  t="1901" r="1409" b="2981"><region><rect l="886"
  t="1901" r="1409" b="2981"/></region>
<text>
<par lineSpacing="1030">
<line baseline="1934" l="893" t="1907" r="1361"
   b="1936"><formatting
   lang="ItalianStandard">TRADUZIONE
   ITALIANA</formatting></line></par>
</text>
...
<text>
<par align="Center" lineSpacing="1090">
</par>
<par startIndent="2100" lineSpacing="1433">
<line baseline="2737" l="985" t="2699" r="1313"
  b="2746"><formatting lang="ItalianStandard">Che
  brilli più!</formatting></line></par>
...
</text>
</block>
...
```

Figure 2. A page scan and a fragment of an XML file after processing by the FRE CLI utility (the blocks framed on the left are described on the right)

## 5 Conclusions

Despite a lot of achievements, automated recognition of the heterogeneous content remains a difficult problem. We proposed a design of

Web platform to maximize the support of semi-automated work with all available tools for recognition of heterogeneous documents.

For implementation, the technology of assembling complex software systems from ready-made modules on a single platform is used. Each of modules performs a small part of the task inside its container. Python's rich libraries dedicated to natural language processing include solutions to the segmentation problem, but without identifying the type of "image". Specifically, only certain types of heterogeneous content are identified, for example, text vs. musical notes. The problem of segmentation, labeling (establishing coordinates), and complete classification of the type of heterogeneous content is not solved completely yet both in scientific publications and in the existing software.

**Acknowledgement.** This article was written as part of the research project 20.80009.5007.22 "Intelligent information systems for solving ill-structured problems, processing knowledge and big data".

## References

[1] B. C. Lee, J. Mears, E. Jakeway, M. M. Ferriter, C. Adams, N. Yarasavage, D. Thomas, K. Zwaard, and D. S. Weld, "The Newspaper Navigator Dataset: Extracting And Analyzing Visual Content from 16 Million Historic Newspaper Pages in Chronicling America". *ArXiv*, abs/2005.01583 (2020).

[2] A. Colesnicov, S. Cojocaru, and L. Malahov, "Recognition of heterogeneous documents: problems and challenges," in *Proceedings of the 5th Conference on Mathematical Foundations of Informatics,* (3-6 July 2019, Iasi, Romania), Iasi: "Alexandru Ion Cuza" University Publishers, 2019, pp. 231–245.ISBN:978-606-714-481-9.

[3] Z. Ziran, X. Pic, S. U. Innocenti, D. Mugnai, and S. Marinai, "Text alignment in early printed books combining deep learning and dynamic programming," *Pattern Recognition Letters*, vol. 133, pp. 109–115, 2020.

[4] W. Qin, R. I. Elanwar, and M. Betke, "LABA: Logical Layout Analysis of Book Page Images in Arabic Using Multiple Support Vector Machines," in *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)*, 2018, pp. 35–40.

[5] S. Chadha, S. Mittal, and V. Singhal, "An Insight of Script Text Extraction Performance using Machine Learning Techniques," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, no. 1, November 2019. ISSN: 2278-3075.

[6] M. Alberti, M. Bouillon, R. Ingold, and M. Liwicki, "Open Evaluation Tool for Layout Analysis of Document Images," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, (Kyoto), 2017, pp. 43–47.

[7] P. Li, X. Jiang, and H. Shatkay, "Extracting Figures and Captions from Scientific Publications," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018.

[8] C. Clausner and A. Antonacopoulos, "Ontology and Framework for Semantic Labelling of Document Data and Software Methods," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, (Vienna), 2018, pp. 73–78.

[9] L. Ma, C. Long, L. Duan, X. Zhang, Y. Li, and Q. Zhao, "Segmentation and Recognition for Historical Tibetan Document Images," in *IEEE Access,* 2020, vol. 8, pp. 52641–52651.

[10] N. Siegel et al., "Extracting Scientific Figures with Distantly Supervised Neural Networks," in *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries*, 2018.

[11] H. M. Al-Barhamtoshy and A. S. Alghamdi, "A Comprehensive Framework for OCR Web Services System for Arabic Calligraphy Documents," *International Journal of Engineering and Technology*, vol. 8, No. 1.11 (2019). DOI: 10.14419/ijet.v8i1.11.28084.

[12] A. Colesnicov, S. Cojocaru, and L. Malahov, "On Digitization of Documents with Script Presentable Content," in *Proceedings of the 5th Conference on Mathematical Society of the Rep. Moldova IMCS-55,* (Chisinau), September 28 – October 01, 2019, pp. 321–324. ISBN: 978-9975-68-378-4.

[13] I. V. Safonov, I. V. Kurilin, M. N. Rychagov, and E. V. Tolstaya, *Document Image Processing for Scanning and Printing*, Springer, 2019, 314 p.

[14] S. S. Kumar, P. Rajendran, P. Prabaharan, and K. P. Soman, "Text/Image Region Separation for Document Layout Detection of Old Document Images Using Non-linear Diffusion and Level Set," *Procedia Computer Science*, vol. 93, pp. 469–477, 2016. ISSN 1877-0509.

[15] P. Sojka, V. Novotný, E. F. Ayetiran, D. Lupták, and M. Štefánik, "Quo Vadis, Math Information Retrieval," in *Proceedings of the Thirteenth Workshop on Recent Advances in Slavonic Natural Language Processing, RASLAN 2019.* Brno: Tribun EU, 2019, pp. 117–128. ISBN 978-80-263-1517-9.

[16] A. M. Namboodiri and A. K. Jain, "Document structure and layout analysis," in *Digital Document Processing*, Springer, 2007, pp. 29–48.

[17] Yu. A. Bolotova, V. G. Spitsyn, and P. M. Osina, "A Review of Algorithms for Text Detection in Images and Videos," *Компьютерная оптика (Computational Optics)*, vol. 41, no. 3, pp. 441–452, 2017.

[18] M. Polyakova, A. Ishchenko, N. Volkova, and O. Pavlov, "Combined Method For Scanned Documents Images Segmentation Using Sequential Extraction of Regions," *Eastern-European Journal of Enterprise Technologies*, vol. 5/2 (95), 2018. ISSN 1729-3774.

[19] S. F. Rashid, A. Akmal, M. Adnan, A. A. Aslam, and A. Dengel, "Table Recognition in Heterogeneous Documents Using Machine

Learning," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, vol. 1*, 2018, pp. 777-–782.

[20] J. Staker, K. Marshall, R. Abel, and C. McQuaw, "Molecular Structure Extraction From Documents Using Deep Learning," [Online]. Available: https://www.x-mol.com/paper/5382953.

[21] B. Kiessling, D. S. B. Ezra, and M. T. Miller, "BADAM: A Public Dataset for Baseline Detection in Arabic-script Manuscripts," in *HIP'19: Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*, 2019, pp. 13-–18.

[22] S. Capobianco, and S. Marinai, "DocEmul: A Toolkit to Generate Structured Historical Documents," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, (Kyoto), 2017, pp. 1186–1191.

[23] F. D. Julca-Aguilar, A. L. L. M. Maia, and N. S. T. Hirata, "Text/Non-Text Classification of Connected Components in Document Images," in *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, (Niteroi), 2017, pp. 450–455. DOI: 10.1109/SIBGRAPI.2017.66.

[24] P. Li, X. Jiang, and H. Shatkay, "Figure and caption extraction from biomedical documents," *Bioinformatics,* vol. 35, no. 21, pp. 4381–4388, November 2019.

[25] C. Roullet, D. Fredrick, J. Gauch, and R. Vennarucci, "An Automated Technique to Recognize and Extract Images from Scanned Archaeological Documents," in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, (Sydney, Australia), 2019, pp. 20–25.

[26] A. S. Ashour and F. Shi, *Machine Learning in Bio-Signal Analysis and Diagnostic Imaging,* Academic Press, 2019, pp. 247–272. ISBN 9780128160862.

[27] N. L. Vine, M. Zeigenfuse, and M. Rowan, "Extracting Tables from Documents using Conditional Generative Adversarial Networks and Genetic Algorithms," in *2019 International Joint Conference on Neural Networks (IJCNN)*, (Budapest, Hungary), 2019, pp. 1–8.

[28] S. Paliwal, D. Vishwanath, R. Rahul, M. Sharma, and L. Vig, "TableNet: Deep Learning Model for End-to-end Table Detection and Tabular Data Extraction from Scanned Document Images," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 128–133.

[29] H. Lane, H. Hapke, and C. Howard, *Natural Language Processing in Action: Understanding, Analysing and Generating Text with Python*, March 2019, 544 p. ISBN: 9781617294631.

[30] D. Chopra, N. Joshi, and I. Mathur, *Mastering Natural Language Processing with Python*, Packt Publishing, June 2016, 237 p.

A. Colesnicov[1,2], L. Malahov[1,3],
S. Cojocaru[1,4], L. Burtseva[1,5]

Received November 05, 2020

[1] "Vladimir Andrunachievici" Institute of Mathematics and Computer Science
5 Academiei str., MD-2028, Chisinau
Republic of Moldova
[2] E–mail: acolesnicov@gmx.com
[3] E–mail: ludmila.malahov@math.md
[4] E–mail: svetlana.cojocaru@math.md
[5] E–mail: luburtseva@gmail.com