Velibor Ilić/Lenka Bajčetić/Snežana Petrović/Ana Španović

SCyDia – OCR FOR SERBIAN CYRILLIC WITH DIACRITICS

Abstract In the currently ongoing process of retro-digitization of Serbian dialectal dictionaries, the biggest obstacle is the lack of machine-readable versions of paper editions. Therefore, one essential step is needed before venturing into the dictionary-making process in the digital environment – OCRing the pages with the highest possible accuracy. Successful retro-digitization of Serbian dialectal dictionaries, currently in progress, has shown a dire need for one basic yet necessary step, lacking until now – OCRing the pages with the highest possible accuracy. OCR processing is not a new technology, as many open-source and commercial software solutions can reliably convert scanned images of paper documents into digital documents. Available software solutions are usually efficient enough to process scanned contracts, invoices, financial statements, newspapers, and books. In cases where it is necessary to process documents that contain accented text and precisely extract each character with diacritics, such software solutions are not efficient enough. This paper presents the OCR software called "SCyDia", developed to overcome this issue. We demonstrate the organizational structure of the OCR software "SCyDia" and the first results. The "SCyDia" is a web-based software solution that relies on the open-source software "Tesseract" in the background. "SCyDia" also contains a module for semi-automatic text correction. We have already processed over 15,000 pages, 13 dialectal dictionaries, and five dialectal monographs. At this point in our project, we have analyzed the accuracy of the "SCyDia" by processing 13 dialectal dictionaries. The results were analyzed manually by an expert who examined a number of randomly selected pages from each dictionary. The preliminary results show great promise, spanning from 97.19% to 99.87%.

Keywords OCR; Cyrillic; Serbian language; retro-digitization; convolutional neural networks

1. Introduction

In the Institute for the Serbian language of SASA, several lexicographic projects – descriptive, etymological, historical, dialectal, neological, etc. – are currently ongoing and still compiled in the traditional way. The lexical material they are based upon includes numerous dictionaries and scientific monographs, which have to be consulted in the paper edition. The vast majority of these dictionaries and monographs (tens of thousands of pages), dedicated to compiling and analyzing dialectal lexis, and describing dialectal features, are written in Cyrillic, containing accents, diacritics, and other non-standard characters. We should bear in mind that the Serbian language is in the position of being low-resourced in the field of digital infrastructure and digitized language resources (for example, in the Institute, no dictionary is corpus-based nor corpus-driven, and no tools for writing or editing dictionaries in the digital environment are used, etc.). Even though some serious first steps have been taken towards applying new technologies to our lexicographic legacy[1] and into the dictionary-making process,[2] we were well aware that this obsolete methodology may question the relevance of research results and downgrade the scientific level of publications. Therefore,

[1] See dictionary platforms Raskovnik and Prepis.

[2] Certain significant steps have been taken also towards digitization of the *Dictionary of the Serbo-Croatian Standard and Vernacular Language of the Serbian Academy of Sciences and Arts* (Stijović/Stanković 2018). Some volumes passed the OCR processing, and manual correction afterwards. However, there is no data on OCR output precision, or how many working hours were spent on corrections (Stanković et al. 2018, p. 942).

This paper is part of the publication: Klosa-Kückelhaus, Annette/Engelberg, Stefan/Möhrs, Christine/Storjohann, Petra (eds.) (2022): Dictionaries and Society. Proceedings of the XX EURALEX International Congress. Mannheim: IDS-Verlag.

we decided to take a broader approach to improve our work – to retro-digitize this vast number of scientific dictionaries and monograph studies of fundamental importance for lexicographic work. That will enable us to create a multifunctional lexicographic database and different corpora and use dialectal material to produce various dictionaries, scientific papers, etc. One of the significant accomplishments of this process of retro-digitization, in the long run, should also be the promotion of dialects and vernaculars, especially in modern-day society. However, the biggest obstacle when attempting to retro-digitize Serbian dialectal dictionaries was the lack of machine-readable versions of paper editions, implying that we needed to complete one essential step before venturing into the dictionary-making process in the digital environment – OCRing the pages with the highest possible accuracy.

Optical Character Recognition (OCR) is a process that allows data extraction from a scanned document or image file. In this process, the printed or handwritten text on the scanned document is converted to a machine-readable format. OCR processing is not a new technology, and there are many open-source and commercial software solutions that can reliably convert scanned images of paper documents into digital documents. Even so, available software solutions are usually efficient enough to process scanned contracts, invoices, financial statements, newspapers, and books. In cases where it is necessary to process documents containing accented text and precisely extract each character with diacritics, such as dialectal dictionaries written with Cyrillic letters, such software solutions are not efficient enough.

1.1 Why OCR?

Although double-keying is the most accurate way for transcription, it is very time-consuming and – in the case of dialectal and historical dictionaries, with text too complex for non-experts – costly because it requires additional corrections, usually more than one. This is based on our previous work experiences digitizing five dialectal dictionaries currently available on *Raskovnik*. Therefore, to overcome this problem, we decided to invest in developing an OCR software called "SCyDia" – *Serbian Cyrillic with Diacritics*. By now, we ran the "SCyDia" software on 14 dictionaries and monographs with more than 15,000 pages combined, but we intend to use it on hundreds of thousands of pages more.

Since the accuracy of OCR varies from 97,19% to 99,87%, some dictionaries would be reasonably quick to verify manually. On the other hand, the worst result of a 2,81% error rate in one dictionary means that a page of 3000 characters has 84,3 errors which can be time-consuming and too expensive to correct. We have opted for a less-than-perfect gradual approach in these cases by correcting only the headword lemmas[3] in the first phase. In this way, we could make our database "searchable" while still keeping the cost reasonably low.

1.2 Related Work

Klyshinsky/Karpi/Bondarenko (2020) compares neural network software used to restore diacritics in six languages such as Croatian, Slovak, Romanian, French, German, Latvian, and

[3] The objective to have a fully and precisely corrected version of the digitized material in Cyrillic with diacritics and other non-standard characters prior to start using it in a lexicographic work process is utmost time-consuming and unrealistic from the financial perspective. See for example Vitas/Krstev (2015, p. 109).

Turkish. The recognition accuracy usually ranges from 95 to 99%, depending on the letter; some letters have relatively low accuracy.

Hussain et al. (2014) present the results of using the Tesseract engine for OCR processing of pages written by Urdu Nastalique (a very complex and cursive writing style of Arabic script); without any modifications, the Tesseract achieves an accuracy of 66%, and with additional modifications, the accuracy is increased to 97%.

Cristea et al. (2020) present the results of a solution based on several types of neural networks (such as The Region Proposal Network (RPN), ResNet, Faster R-CNN) for OCR processing of old Romanian documents written in Cyrillic.

Rijhwani/Anastasopoulos/Neubig (2020) describes post-correction methods where the goal is to reduce the number of errors that occur during OCR processing that most often happen due to low-quality scanning, physical deterioration of paper book, or different styles of font.

In their research, Krstev/Stanković/Vitas (2018) present the process of restoring diacritics in Serbian texts written in degraded Latin script, and the presented solution relies on the comprehensive lexical resources for Serbian: the morphological electronic dictionaries, the Corpus of Contemporary Serbian and local grammars.

In their research, O'Brien/Haddej (2012) present a project where the functionality of OCRopus software has been expanded to support the recognition of mathematical symbols and unique linguistic alphabets (e. g., Hungarian letters) while the extended version supports UTF-8 character encoding. The accuracy of the original version trained only with English characters was 86%; in the extended version, the accuracy increased to 93,5%.

1.3 An overview of the "SCyDia" software

This paper will present the OCR software "SCyDia", a web-based software solution that relies on open-source software Tesseract V5 in the background. The software is developed to overcome the problem of not having OCR software efficient enough to process documents containing accented text and precisely extract each character with diacritics. Finally, we will demonstrate the organizational structure of the software and the first results.

The paper is organized as follows. Section 2 contains implementation details, details about used convolutional neural networks (CNN) and datasets, and a description of modules for semi-automatic text correction. After that, section 3 presents the results. Further plans are presented in section 4. Finally, the last section contains conclusions.

2. Implementation of SCyDia

The "SCyDia" OCR software is developed as a web application; an overview of the algorithm is presented in Figure 1. It allows the user to see a list of scanned pages and select pages for OCR pressing or text correction (proofreading).

The web application (1) allows the user to choose which scanned pages will be processed. The selected images of the scanned text pages (2) are forwarded to the Python application. OCR processing in the initial step uses Tesseract (3), which generates a text file (6) with recognized text without diacritic signs. Tesseract also returns coordinates of bounding boxes around individual letters. The coordinates of bounding boxes are usually concretely determined. Occasionally, instead of one letter inside the bounding box, it may contain two, three, or even more letters; sometimes, the bounding box can contain halves of two adjacent letters.

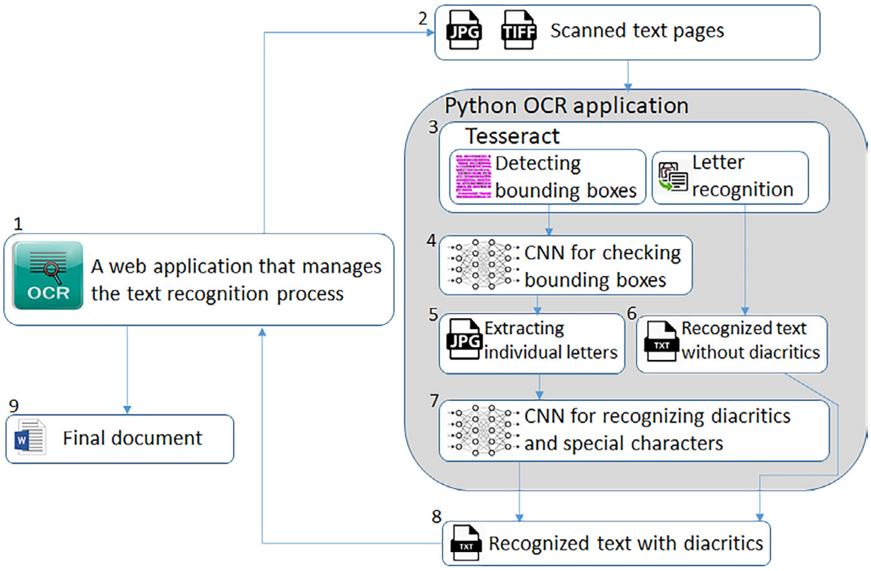
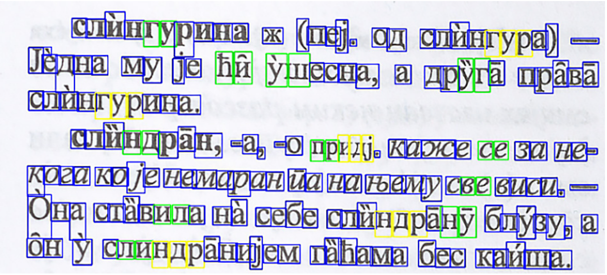

2
Scanned text pages
Python OCR application
3
Tesseract
Detecting bounding boxes
Letter recognition
4
CNN for checking bounding boxes
5
Extracting individual letters
6
Recognized text without diacritics
7
CNN for recognizing diacritics and special characters
1
OCR
A web application that manages the text recognition process
9
Final document
8
Recognized text with diacritics
JPG
TIFF
TXT

Fig. 1: Overview of ScyDia software

The convolutional neural network (4) can check whether the bounding box contains only one letter as expected, and if there is more than one, it returns information on how many letters are inside the bounding box. For example, detected bounding boxes with more than one letter are divided into an appropriate number of smaller bounding boxes containing one letter.

In Figure 2, the correctly determined bounding boxes with one letter are shown in blue. Those boxes that initially contained two letters and were divided into two parts are shown in green, and boxes with three letters are divided into smaller boxes are shown in yellow color. Bounding Border boxes where multiple letters are detected are automatically divided into the appropriate number of parts to contain one letter using the Python script.



сли̏нгурина ж (пеј. од сли̏нгура) — Јѐдна му је ћи̂ у̀шесна, а дру̏га̄ пра̂ва̄ сли̏нгурина.

сли̏ндра̄н, -а, -о прид. *каже се за некога ко је немаран па на њему све виси.* — О̏на ста̏вила на̀ себе сли̏ндра̄нӯ блу́зу, а о̂н у̀ сли̏ндра̄нијем га̏ћама бес каи́ша.

Fig. 2: Detected bounding boxes around letters

In Figure 1, Python script (5) uses bounding boxes coordinates to extract individual letters' images. The convolutional network (7) processes those images of individual letters and tries to detect whether they contain diacritic signs. Also, this network can be used to detect letters that Tesseract has difficulty recognizing correctly, such as italic letters *ī ū ш̄ в*. In the final step, the Python function tries to match each letter from a text file with the information provided by the convolutional network when processing extracted images of those letters. The result of that function represents a new text file containing letters with diacritic signs. For example, “SCyDia” software generates text in format UTF-8 plaintext; letters with diacritics consist of two characters, one character for the letter and the other for the diacritical character (symbol).

2.1 Network Configuration and Datasets

The “SCyDia” OCR application uses two convolutional neural networks, CNN for checking bounding boxes and CNN for detecting diacritics. These two networks have similar configurations, and they differ in the number of outputs.

The **CNN used for detecting diacritics** takes a 48×32×1 matrix as input; it contains three convolutional layers. The first layer contains 16, the second 32, and the third layer contains 64 3×3 kernels with *ReLu* activation. After each layer, a max-pooling layer with a pooling size of 2×2, a dropout probability of 0.25 is placed. Three fully connected layers follow these convolutional layers: the first layer contains 128 nodes, the second 64 nodes, and the third layer contains 32 output neurons. After each layer is placed, the dropout layer with a dropout probability of 0.25. Finally, the output layer contains 30 nodes, Figure 3. The values obtained at the network output have the following meaning: the first value indicates whether the letter contains diacritic signs, the second whether the letter is correct (sometimes the bounding box is not placed correctly around the letter), and the following 15 values detect the type of diacritic signs, the remaining values are used to detect letters Tesseract does not recognize correctly, for example, letters (ьəѣŋʒ, and italic letters such as *ī ū ш̄ в*).

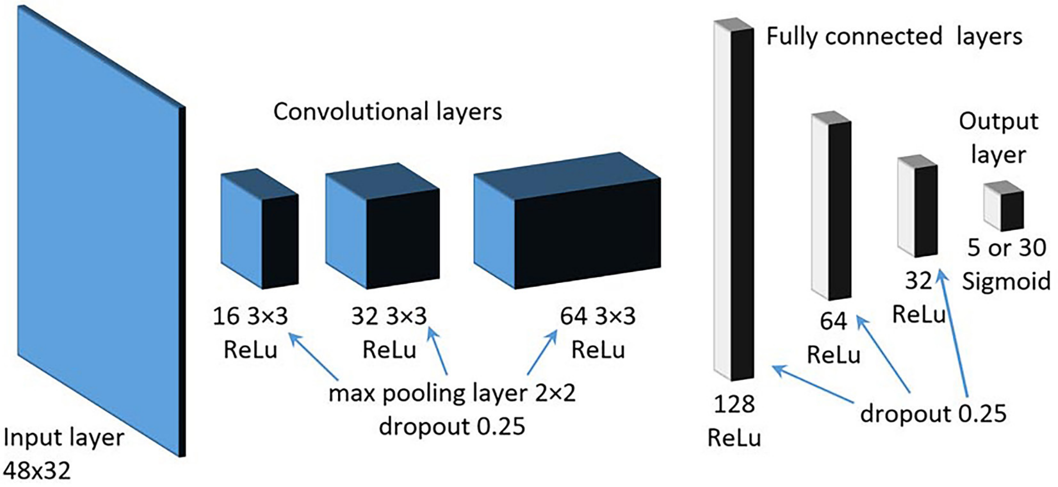


Fig. 3: Structure of convolutional networks

Datasets for CNN used for detecting diacritics are generated by collecting cropped individual letters from scanned pages. This dataset contains:[4]

[4] It’s worthwhile noting that all scholar dictionaries in Serbian, and even most of the popular ones, are using characters with diacritics.

Group of Cyrillic letters:

- Standard set of Cyrillic letters,
- Letters that have diacritics above the letters, for example:

  а̀ а́ а̏ а̂ а̄ а̍ ӑ а̃ а̇́ ӑ̌ а̇
- Letters that have diacritics below the letters, for example:

  а̣ а̤ а̨л̥
- Letters that have diacritics above and below the letters,
- Cyrillic letters that do not belong to the standard set of symbols that Tesseract cannot recognize, for example:

  ьәѣŋʒ Tesseract incorrectly recognizes these letters as: Б о ђ о з
- Letters where one letter consists of two characters, for example: д͡з

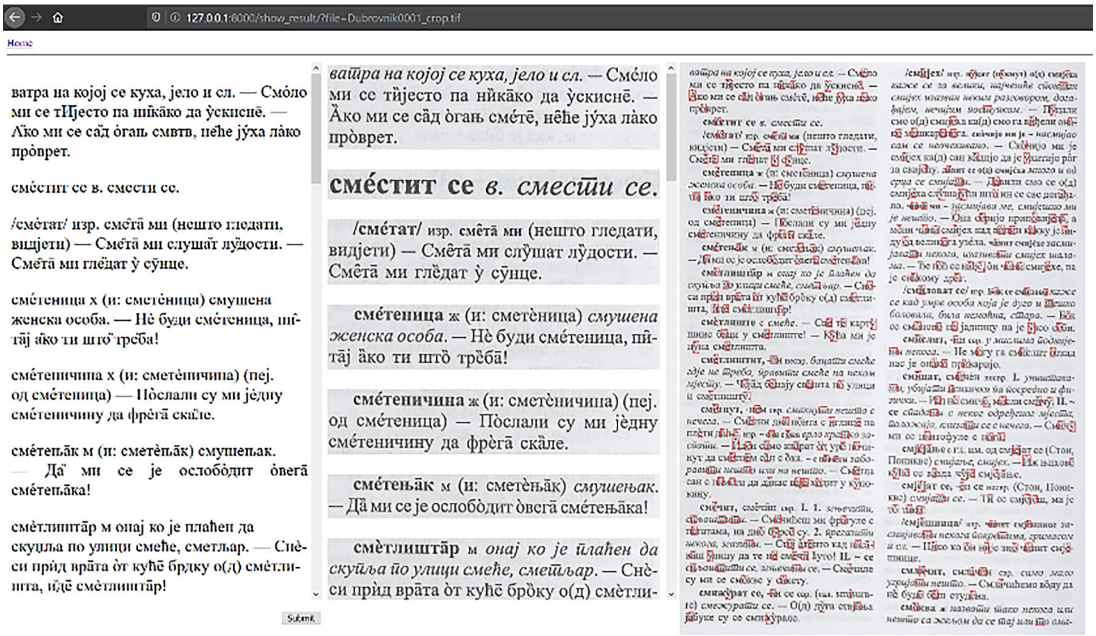
The **CNN used for checking bounding boxes** has a similar configuration; the output layer of that network contains 5 nodes, Figure 3. The values obtained at the network output have the following meaning – the first value indicates that bounding box is around one letter, and the second value indicates that bounding box is around two letters. The third value indicates that the bounding box is around three letters, the fourth value indicates more than three letters, and the fifth value is used to detect invalid letters; for example, there are two halves of consecutive letters within the boundary frame.

Datasets for CNN used for checking bounding boxes are also generated by collecting cropped letters from scanned pages. This dataset contains examples of how an adequately extracted letter looks, examples of when two or three letters are extracted together, and examples of images with incorrectly extracted letters when two halves of a letter are in a boundary field.

Adam optimizer is used for both networks. The duration of training was limited to 50 epochs, with two additional parameters: *ReduceLROnPlateau* with patience 10 and *EarlyStopping* with patience 25. Parameter *ReduceLROnPlateau* would reduce the learning rate if there were no improvement in the accuracy of the validation dataset for 10 epochs. *EarlyStopping* interrupts training if there is no improvement in the accuracy of the validation dataset for 25 epochs.

2.2 Manual and semi-automatic text correction (proofreading)

The primary purpose of the “SCyDia” application is OCR processing; besides that web application also provides a module for text correction (proofreading). That module allows manual and semi-automatic text correction. The window for **manual text correction** is divided into three fields (Fig. 4), the first field contains the recognized text, and it is an editable field; the second field contains cut-out images of paragraphs; in the third field, there is a complete picture of the scanned page on which the letters containing diacritics are marked.



ватра на којој се куха, јело и сл. — Смо́ло ми се тИјесто па ни̏ка̄ко да у̏ксисне̄. — А̏ко ми се са̏д о̀гањ смвтв, не̄ће ју́ха ла̀ко про̀врет.

сме́стит се в. смести се.

/сме́тат/ изр. сме̑та̄ ми (нешто гледати, виђјети) — Сме̑та̄ ми слушат лу̏дости. — Сме̑та̄ ми глёдат у̏ су̑нце.

сме́теница х (и: смете̇ница) смушена женска особа. — Нѐ буди сме́теница, пи̏та̄ј а̏ко ти што̏ тре̑ба!

сме́теничина х (и: смете̇ничина) (пеј. од сме́теница) — По̀слали су ми је̏дну сме́теничину да фрѐга̄ скаље.

сме́теља̄к м (и: сметѐља̄к) смушењак. — Да̏ ми се је ослобо̏дит о̀вега̄ сме́теља̄ка!

смѐтлиштта̄р м онај ко је плаћен да скуцља по улици смеће, сметљар. — Снѐси при̏д вра̄та о̀т куће̄ брдку о(д) смѐтлишта, и̏де̄ смѐтлишта̄р!

Fig. 4: Window for manual text correction

In order to **achieve semi-automatic text correction** (Fig. 5), the "SCyDia" application repeats OCR processing (3) of one page several times to create additional copies of text files that can be compared with each other. The algorithm for semi-automatic text correction starts by creating additional two copies (2) of the scanned page (1), the first image is rotated to the left by half a degree, and the second copy is rotated to the right half a degree. If they visually compare those images, humans will not notice the differences between the original scanned page and copies of that image rotated by half a degree. However, for OCR software, such a small difference causes misrecognized letters to appear in different places in the recognized text.

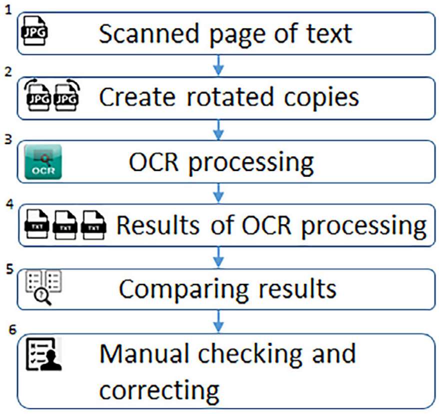
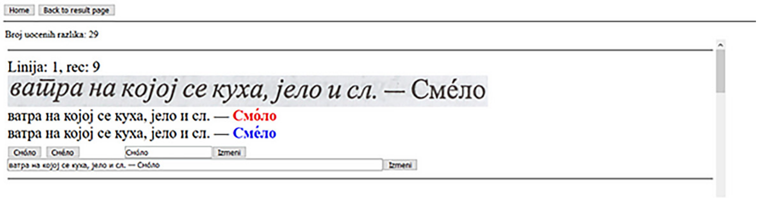


Fig. 5: Algorithm for semi-automatic text correction

In the next step, those text files (4) are compared with each other (5), and each detected difference is presented on the window for manual checking and correcting (6). In most cases, users can click on the button with the correct version of a word, Figure 6.



Broj uocenih razlika: 29

Linija: 1, rec: 9

ваӣра на којој се куха, јело и сл. — Сме́ло

ватра на којој се куха, јело и сл. — Смо́ло

ватра на којој се куха, јело и сл. — Сме́ло

Смо́ло

ватра на којој се куха, јело и сл. — Смо́ло

Fig. 6: User interface with results of semi-automatic text correction

The user interface with the results of the semi-automatic text correction contains following elements:

- the part of the scanned image with the text line where the difference is noticed,
- the text line where the difference is noticed from the original scanned page (word where the difference is presented in red),
- the text line where the difference is noticed from the rotated copy (word where the difference is presented in red),
- Button with a version of word from the first file,
- Button with a version of word from the second file,
- A text box that allows the user to manually correct an error if neither of these two versions is correct.

2.3 Usage of “SCyDia”

The “SCyDia” application has so far been used for processing over 15 000 pages of dialectical dictionaries of Serbian. The OCR process is conducted on a PC with Intel I9 12-core processor, with NVidia GeForce RTX 2070 SUPER graphic card. The “SCyDia” application can process eight pages in parallel, and each page is analyzed three times: first in its original shape and then skewed for half a degree left and right. On average, each page takes about half an hour to process. After the first batch of 14 dictionaries was processed, the results were analyzed. We have compiled a list of the most common problems for each dictionary. A list of letters and diacritics signs has been compiled, with the most common problems in each dictionary. Based on this list, an additional set of images with letters and diacritics will be generated to expand the dataset for training CNN used to detect diacritics.

3. Results

3.1 General characteristics of processed dictionaries

Table 1 provides an overall description of 13 dictionaries processed in the “SCyDia” application by showing some of their main characteristics relevant for the OCR, such as the possession of characters with diacritics in the headword, characters with diacritics in the citation, characters in italic, abbreviations, as well as characters in superscript.

The accuracy of OCR processing is evaluated by comparing the text generated by the OCR software with the reference text (manually typed text); the comparison is performed using a script, and the results obtained are shown in the following table.

DICTIONARIES	CHARACTERS WITH DIACRITICS IN THE HEADWORD	CHARACTERS WITH DIACRITICS IN CITATION	CHARACTERS IN CURSIVE	ABBREVIATIONS	SUPERSCRIPT
Bašanović-Čečović (2010)	+	+	+	+	+
Boričić Tivranski (2002)	+	-	+	+	-
Bukumirić (2012)	+	+	+	+	-
Cvetanović (2013)	+	+	-	+	-
Cvijetić (2014)	+	+	+	+	-
Dalmacija (2004)	+	+	+	+	+
Dalmacija (2017)	+	+	+	+	+
Đoković (2010)	+	-	-	+	-
Rajković Koželjac (2014)	+	+	+	+	-
Ristić (2010)	+	+	+	+	+
RSGV (2000–)	+	+	+	+	-
Stanić (1990–1991)	+	+	+	+	+
Zlatković (2014)	+	+	+	+	-

Table 1: Overall description of dictionaries' complexity

As expected, characters with diacritics in the headword are present in each of the 13 dictionaries. Characters with diacritics in the citation are documented in most dictionaries (11 out of 13), except in Boričić Tivranski (2002) and Đoković (2010). 11 out of 13 dictionaries have characters in cursive, except Cvetanović (2013) and Đoković (2010). Abbreviations, such as grammatical ones, and locations and sources are present in all 13 dictionaries. Finally, superscript is found in 5 out of 13 dictionaries and missing from Boričić Tivranski (2002), Bukumirić (2012), Cvetanović (2013), Cvijetić (2014), Đoković (2010), Rajković Koželjac (2014), RSGV (2000–), and Zlatković (2014).

3.2 OCR processing accuracy

The accuracy of OCR processing was evaluated manually by experts. Although the "SCyDia" software provides semi-automatic detection of errors by comparing the slightly rotated versions to the original, we have decided to evaluate manually to ensure that the evaluation results are as accurate as possible. Semi-automatic error detection is beneficial for manual correction, but we cannot be sure that all errors are detected in this way. The experts counted all errors on the page and errors in "special" characters: letters with diacritics, italic, and specific abbreviations. Finally, we wanted to see to what extent these special characters affect the results of the OCR so we could see what aspects we need to improve.

DICTIONARIES	TN CHARACTERS	TN ERRORS	% CORRECT	TN LETTERS WITH DIACRITICS	TN ERRORS IN DIACRITICS	% CORRECT	% ERRORS IN DIACRITICS VS. TN ERRORS
Cvetanović (2013)	1455	2	99,87	107	/	100	/
Đoković (2010)	2761	4	99,86	/	/	/	/
Boričić Tivranski (2002)	1232	2	99,84	45	/	100	/
Cvijetić (2014)	2791	17	99,39	30	/	100	/
Zlatković (2014)	3422	33	99,04	263	6	97,8	18,18
Stanić (1990–1991)	4394	62	98,59	263	16	94	25,80
Ristić (2010)	2938	43	98,54	312	25	92	58,13
Dalmacija (2017)	2047	30	98,53	193	15	92,2	50
Rajković Koželjac (2014)	3011	47	98,44	175	20	88,6	42,55
Dalmacija (2004)	2938	38	98,42	329	5	98,5	13,15
RSGV (2000–)	3566	79	97,74	161	14	91,3	17,72
Bašanović-Čečović (2010)	2853	61	97,86	355	35	90,1	57,37
Bukumirić (2012)	2563	72	97,19	256	6	97,7	8,33

Table 2: Accuracy of OCR processing

As it is shown in Table 2, three dictionaries have the highest accuracy percentage – 99,87% in Cvetanović (2013), Đoković (2010) and 99,86%, and 99,84% in Boričić Tivranski (2002). A mutual characteristic they all share is zero errors in characters with diacritics. In addition, one more dictionary is processed without errors in diacritics, Cvijetić 2014, making it a total of four.

When it comes to the total number of errors in diacritics, most of them are linked to characters in cursive. Dictionaries that have diacritics in cursive have the most mistakes in diacritics – Rajković Koželjac (2014) with 20 out of 175 total characters with diacritics (88,6% of accuracy), Bašanović Čečović (2010) with 35 out of 355 total (90,1%) and RSGV (2000)– with 14 out of 161 total (91,3%).

A specific type of error in characters with diacritics is present in most dictionaries – the letter o with any sort of diacritic is mistakenly read by the “SCyDia” application as the Cyrillic letter д. The most significant number of these errors is found in two dictionaries (Ristić 2010; Dalmacija 2017), where they form more than 50% of all errors in characters with diacritics.

DICTIONARIES	TN CHARACTERS IN CURSIVE	TN ERRORS IN CURSIVE	TN ABBREVIATIONS	TN ERRORS IN ABBREVIATIONS
Cvetanović (2013)	/	/	75	/
Đoković (2010)	/	/	78	3
Boričić Tivranski (2002)	85	1	55	1
Cvijetić (2014)	798	17	228	5
Zlatković (2014)	755	12	298	6
Stanić (1990–1991)	1252	55	267	4
Ristić (2010)	828	1	75	/
Dalmacija (2017)	669	34	54	2
Rajković Koželjac (2014)	231	16	125	/
Dalmacija (2004)	820	1	83	/
RSGV (2000–)	148	1	452	20
Bašanović-Čečović (2010)	627	2	71	2
Bukumirić (2012)	483	14	152	17

Table 3: Accuracy of OCR processing additional data

Table 3 is providing further results obtained from processing the dictionaries in the “SCyDia” application.

What the results in the table are showing is that the presence (or lack) of cursive is crucial to the total percentage of errors, especially if cursive is combined with diacritics. Dictionaries with the highest percentage of errors (Bašanović Čečović 2010; Dalmacija 2017) have both characters in cursive and with diacritics. Similarly, dictionaries with the highest percentage of accuracy, such as Đoković (2010), Cvetanović (2013) don’t have characters in cursive.

These results are similar to ones obtained by Polomac and Lutovac Kaznovac in their work with OCR for Serbian medieval manuscripts: “An extraordinarily high percentage of errors indicates that it is necessary to train a separate model for the automatic recognition of manuscripts written in cursive script” (Polomac/Lutovac Kaznovac 2021, p. 16). Although their system is trained to recognize manuscripts and Old Slavonic letters, it is interesting to see that cursive poses the biggest problem similarly to our results. It is also noteworthy to point out that the significan percentage of errors in their research are most frequently related to the blanks between words, superscript letters and titles, i. e. diacritics (ibid., pp. 23 f.).

4. Further plans

Once the transcribed text is manually corrected, we will place results in structured dictionary. We are currently developing an OntoLex schema that would be suitable for all the dictionaries and enable the smooth integration of various resources into one connected data structure. In the end, we want to create a web app with which some parts of the database would be accessible to the broader public, and some would require a license to access, depending on the copyright of the dictionary. Also, the web app would allow a certain number of users to edit mistakes that may have remained after OCR and the scarce manual correction.

5. Conclusions

Today, when most dictionaries are being produced in digital form, it is essential not to lose sight of those that, for now, exist in paper form only and need to be transformed into a digital, computer-readable format. Breathing new life into non-digital lexicographic works requires a lengthy, multi-step process of retro-digitization. The end goal is to produce structured and indexed material that can be searched and integrated into various lexicographic projects, from scholarly dictionaries to more popular content. Still, in the case of the Serbian language, this end goal may look out of reach until some basic requirements are fulfilled. The presented "SCyDia" software solution is just one – but vital – step towards building up-to-date, multipurpose, and scientifically reliable digital linguistic resources for Serbian. "SCyDia" is developed as open-source software is available and it is available on GitHub at the following link: https://github.com/ilicv/Cyrilic_OCR.

References

Bašanović-Čečović, J. (2010): Rječnik govora Zete. Podgorica. [Vocabulary of Zeta (in Cyrillic)]

Boričić Tivranski, V. (2002): Rječnik vasojevićkog govora. Beograd. [Vocabulary of Vasojevići (in Cyrillic)]

Bukumirić, M. (2012): Rečnik govora severne Metohije. Beograd. [Dictionary of the north Metohia (in Cyrillic)]

Cristea, D./Pădurariu, C./Rebeja, P./Onofrei, M. (2020): From scan to text. Methodology, solutions and perspectives of deciphering old cyrillic Romanian documents into the Latin script. In: Knowledge, Language, Models, pp. 38–56. https://profs.info.uaic.ro/~dcristea/papers/Paper%20volume%20Bulgaria-Cristea_etAl.pdf (last access: 15-03-2022)

Cvetanović, V. (2013): Rečnik zaplanjskog govora. Gadžin Han. [Vocabulary of Zaplanje (in Cyrillic)]

Cvijetić, R. (2014): Rečnik užičkog govora. Užice. [Vocabulary of Užice (in Cyrillic)]

Dalmacija, S. (2004): Rječnik govora Potkozarja. Banja Luka. [Vocabulary of Potkozarje (in Cyrillic)]

Dalmacija, S. (2017): Rječnik govora Srba zapadne Bosne. Banja Luka. [Vocabulary of Serbian vernaculars of western Bosnia (in Cyrillic)]

Đoković, Lj. (2010): Rječnik nikšićkog kraja. Podgorica. [Vocabulary of the area of Nikšić (in Cyrillic)]

Hussain, S./Niazi, A./Anjum, U./Irfan, F. (2014): Adapting tesseract for complex scripts: An example for Urdu Nastalique. In 2014 11th IAPR International Workshop on Document Analysis Systems. IEEE, pp. 191–195.

Klyshinsky, E./Karpik, O./Bondarenko, A. (2020): A comparison of neural networks architectures for diacritics restoration. In: Recent Trends in Analysis of Images, Social Networks and Texts. AIST 2020. Communications in Computer and Information Science 1357, pp. 242–253.

Krstev, C./Stankovic, R./Vitas, D. (2018): Knowledge and rule-based diacritic restoration in Serbian. In: Computational Linguistics in Bulgaria. Proceedings of the Third International Conference (CLIB), Sofia, 28–29 May 2018. Sofia, pp. 41–51.

O'Brien, S./Haddej, D. B. (2012): Optical character recognition. Degree of Bachelor of Science. Worcester Polytechnic Institute.

Polomac, V./Lutovac Kaznovac, T. (2021): Automatic recognition of Serbian medieval manuscripts by applying the transcribus software platform: Current situation and future perspectives. In: Zbornik Matice srpske za filologiju i lingvistiku 64/2, pp. 7–26.

Prepis: http://www.prepis.org/ (last access: 01-03-2022).

Rajković Koželjac, Lj. (2014): Rečnik timočkog govora. Negotin. [Vocabulary of Timok (in Cyrillic)]

Raskovnik: http://raskovnik.org/ (last access: 01-03-2022).

RSGV (2000–): Rečnik srpskih govora Vojvodine. Novi Sad. [Vocabulary of Serbian vernaculars of Vojvodina (in Cyrillic)]

Rijhwani, S./Anastasopoulos, A./Neubig, G. (2020): OCR post correction for endangered language texts. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, pp. 5391–5942. https://aclanthology.org/2020.emnlp-main.478/ (last access: 15-03-2022)

Ristić, D. (2010): Rječnik govora okoline Mojkovca. Podgorica. [Vocabulary of the area of Mojkovac (in Cyrillic)]

Stanić, M. (1990–1991): Uskočki rečnik. Beograd. [Vocabulary of Uskoci (in Cyrillic)]

Stanković, R./Stijović, R./Vitas, D./Krstev, C./Sabo, O. (2018): The dictionary of the Serbian Academy: From the text to the lexical database. In: Lexicography in global contexts. Proceedings of the 18th EURALEX International congress, Ljubljana, 17–21 July. Ljubljana, pp. 941–949. https://euralex.org/publications/the-dictionary-of-the-serbian-academy-from-the-text-to-the-lexical-database/ (last access: 05-03-2022).

Stijović, R./Stanković, R. (2018): Digitalno izdanje Rečnika SANU: formalni opis mikrostrukture Rečnika SANU. In: Naučni sastanak slavista u Vukove dane 47/1, pp. 427–440.

Vitas, D./Krstev, C. (2015): Nacrt za informatizovani rečnik srpskog jezika. In: Naučni sastanak slavista u Vukove dane – Srpski jezik i njegovi resursi: teorija, opis i primene 44/3, pp. 105–116. [Blueprint for the computerized dictionary of the Serbian language (in Cyrillic)]

Zlatković, D. (2014): Rečnik pirotskog govora. Beograd. [Vocabulary of Pirot (in Cyrillic)]

Contact information

Velibor Ilić
The Institute for Artificial Intelligence Research and Development of Serbia, 21000 Novi Sad, Serbia
velibor.ilic@ivi.ac.rs

Lenka Bajčetić
Institute for the Serbian Language of SASA
lenka.bajcetic@gmail.com

Snežana Petrović
Institute for the Serbian Language of SASA
snezzanaa@gmail.com

Ana Španović
Institute for the Serbian Language of SASA
tesicana@gmail.com

Acknowledgements

This paper was funded by the Ministry of Education, Science and Technological Development of the Republic of Serbia according to the Agreement No. 451-03-68/2022-14 concluded with the Institute for the Serbian Language of SASA.