Algorithms for Optical Character Recognition of Ukrainian Printed Editions of the Late Nineteenth and Early Twentieth Centuries*

Volodymyr Yarych[1,*,†] and Grygoriy Melnyk[1,†]

[1] West Ukrainian National University, Lvivska str. 11, 46000 Ternopil, Ukraine

Abstract

This paper investigates optical character recognition of Ukrainian printed editions of the late nineteenth and early twentieth centuries using Tesseract 5. The study focuses on documents with historical orthography, mixed-language fragments, old typefaces, and typical scan defects such as broken strokes, ink spread, and background interference. A dataset of book and periodical sources was prepared and manually segmented to reduce layout-related errors and isolate the recognition stage. We first evaluated the standard Ukrainian Tesseract model and then applied fine-tuning with an expanded character set and synthetic line images containing historically relevant graphemes and contrastive samples. Recognition quality was assessed using Character Error Rate and Word Error Rate. The baseline model showed acceptable performance only on fragments close to modern orthography, while the most difficult source yielded CER = 18.75% and WER = 61.6%. After fine-tuning, the same source improved to CER = 10.95% and WER = 51.3%, confirming that adaptation of the OCR model to historical orthography and typography can substantially improve recognition quality.

Keywords

Historical OCR, Tesseract 5, Historical Source Studies, Fine-Tuning, Charac-ter Error Rate, Word Error Rate, Historical Source Studies

1. Introduction

The development of digital technologies creates new opportunities for preserving, organizing, and studying historical documents. One important area is the use of optical character recognition (OCR) systems, in particular Tesseract 5. This helps expand access to archival materials and simplifies their analysis.

At the same time, historical printed sources are difficult to recognize automatically. They differ in fonts, spelling, page layout, and physical condition. Many documents contain faded text, stains, paper damage, decorative elements, columns, footnotes, or illustrations. These factors reduce OCR accuracy and make text segmentation more difficult.

A separate challenge is the language itself. Ukrainian printed editions of the late nineteenth and early twentieth centuries reflect a period when spelling was not yet standardized. Texts may contain archaic letters, variant spellings, and forms that differ from modern usage. In addition, some sources combine several languages or alphabets on one page. Because most OCR models are trained on modern language data, they often produce more errors on such materials. The study is based on a corpus of Ukrainian printed editions from the late nineteenth and early twentieth centuries.

Standard OCR models often produce errors at the character, word, and document-structure levels, which reduces the usefulness of the results for further search, linguistic analysis, and the creation of digital corpora of historical texts. Therefore, it becomes necessary to investigate the specific features of recognizing such editions with Tesseract, identify the main factors that reduce OCR quality, compile a representative dataset with characteristic examples of old fonts, historical

* The Third International Conference of Young Scientists on Artificial Intelligence for Sustainable Development: Intelligent Systems and Artificial Intelligence (YAISD-WS 2026:ISAI) May 7-8, 2026, Ternopil, Ukraine

[1,*] Corresponding author.

[†] These authors contributed equally.

v.yarych@wunu.edu.ua (V. Yarych); mgm@wunu.edu.ua (G. Melnyk)

0000-0003-4455-952X (V. Yarych); 0000-0003-0646-7448 (G. Melnyk)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-11

CEUR-WS.org/Vol-4326/paper6.pdf

orthography, typographic elements, and scan defects, and evaluate recognition accuracy on this material using formalized metrics. Solving this problem is important for improving the digitization of historical book sources and creating text resources suitable for further automated processing.

The aim of this study is to investigate the OCR quality of historical Ukrainian book editions, identify the key factors that cause recognition errors, and develop a methodology for fine-tuning the Tesseract system to improve the recognition quality of such sources. The main contributions of this paper are as follows:

1. a methodology for adapting the Tesseract 5 Ukrainian OCR model to printed Ukrainian editions of the late nineteenth and early twentieth centuries is proposed,
2. a synthetic line generation procedure is used to create training samples that reflect the historical orthography, obsolete Cyrillic characters, and graphical properties of the target document collection.

2. Related Work

The recognition of historical documents is a separate and challenging task that depends not only on the OCR engine, but also on scan quality, page structure, font characteristics, and paper condition. Studies [1–5] show that for early print-ed books and historical book materials, general OCR approaches without addi-tional adaptation are often not accurate enough. The combination of structure detection and fine-tuned OCR reduced Character Error Rate (CER) by 15.68 per-centage points and Word Error Rate (WER) by 19.95 for a multi-column docu-ment [6]. The authors of [7] show that an effective OCR system for historical documents can be developed even with a small amount of annotated data, and the results were comparable to or better than several state-of-the-art systems.

Tesseract is an open-source OCR tool that can be adapted to specific lan-guages, font systems, and historical sources. Fine-tuning the model or using spe-cially prepared training data in studies [8, 3, 9] significantly reduced the number of errors compared with baseline configurations. Fine-tuning for non-standard fonts [10] reduced CER from 6.03 to 2.61 for Tamil and from 7.61 to 4.74 for Sinhala. The use of mixed training based on a combined dataset and fine-tuning in [2] reduced CER to 1.47%.

Studies [11, 12] examine OCR for Cyrillic script, Slavic languages, and histor-ical sources. The authors of [12] combine Tesseract with post-OCR correction for Slavic Cyrillic, which made it possible to correct up to 88% of incorrectly recog-nized terms. In [13], the authors created a benchmark and a synthetic data pipe-line for historical Bulgarian orthography. They achieved a 25% improvement in document quality, which was 16% better than the previous state of the art.

Article [14] examines the problems of digitizing early printed books. The au-thors argue that the quality of a document's further digital representation is de-termined even before the text recognition stage, namely by the condition of the paper, the nature of printing defects, scanning parameters, lighting, the level of digital noise, and the capabilities of software processing. The authors describe such defects as low contrast, uneven background, transparent and opaque stains, folds, tears, bleed-through from the reverse side, noise, and extraneous marks. TIFF format and a resolution of 600 ppi or higher were used to create digital cop-ies.

OCR accuracy also strongly depends on the correct detection of text blocks, lines, columns, and other page elements [2]. This is especially important for his-torical books, since complex layouts, page damage, decorative elements, and uneven printing may cause errors even before the actual text recognition stage. Study [15] shows that OCR of historical books begins with layout recognition. The authors demonstrate how deep models and background knowledge improve the recognition of text lines and page structure.

The use of OCR technologies is especially important in the field of cultural heritage and for Ukrainian-language texts. Automatic text recognition [16] signif-icantly accelerates the digitization of archival materials, simplifies document search, and expands remote access to sources, which is

particularly important for cultural heritage preservation and genealogical research. The authors of [17] demonstrate that Ukrainian OCR can be integrated into real-time systems for mobile text recognition and speech output. The results of [18] show that OCR effectiveness strongly depends on the type of source and the nature of distor-tions. Therefore, achieving high quality requires careful tool selection, proper configuration, neural network fine-tuning, and image preprocessing.

3. Dataset Description

The selection of sources for the corpus was carried out with regard to the historical, orthographic, and publishing features of Ukrainian printed editions of the nineteenth and early twentieth centuries. These factors directly affect the complexity of OCR recognition and the representativeness of the dataset. In the nineteenth century, under difficult political conditions, the formation of modern Ukrainian orthography began, and this was reflected in printed texts. Within Ukrainian orthography, four basic principles gradually became established: phonetic, morphological, historical, and differentiating. The importance of the language factor for national revival was emphasized by leading intellectuals of the period [19]. The consolidation of the Ukrainian written word was associated with the intensification of research in ethnography, linguistics, and history as components of national and cultural development [20].

Writers who focused on the living vernacular actively sought ways to represent its phonetic features more accurately. During the nineteenth century, the alphabet system underwent significant changes. In 1818, the letter “і” was introduced; in 1837, the letter “є” and the letter combinations “йо” and “ьо” appeared. In 1873, the letter “ї” was introduced, while the letters “ъ,” “ы,” and “э” gradually fell out of use. The intensity of these transformations led to the emergence of a large number of orthographic systems. From the late eighteenth century to the early twentieth century, there were about fifty of such systems.

Among the best-known orthographic concepts, special attention should be given to the systems of O. Pavlovskyi, M. Maksymovych, P. Kulish, M. Drahomanov, Ye. Zhelekhivskyi, and S. Smal-Stotskyi and F. Gartner (1893). The latter became one of the most consistent implementations of the phonetic principle and introduced specifically Ukrainian terminology.

The development of Ukrainian orthography took place under administrative pressure in the late nineteenth century. Until the beginning of the Ukrainian Revolution of 1917–1921, Ukrainian book publishing remained limited, and some texts were printed using Russian script. As P. Hrytsenko notes, only a small number of grammars and dictionaries were allowed to be published, while a substantial body of nineteenth-century linguistic works remained in archives or existed only in manuscript form [21].

The historical sources were selected so that the corpus would reflect the diversity of orthographic systems, language interference, and typographic practices that create characteristic difficulties for automatic text recognition. Information on the year and place of publication of each document is presented in Table 1. Further in the text, the collection documents will be marked with the corresponding ID.

Table 1.
Dataset.

ID	Title	Year	Place	Color	Pages
PK1876	Pro kary, do yakykh prysudzhuiut myrovi suddi.	1876	Kyiv	color	35
ZNTSH1894	Zapysky tovarystva imeni Shevchenka. T. 4	1894	Lviv	binary	204

KS1904	Kievskaya Starina	1904	Kyiv	binary	248
SV1927-5	Silski visti.	1927	Lviv	color	48

The source ZNTSH1894 is a scholarly publication in which a single paragraph may simultaneously contain Ukrainian, Polish, German, and Church Slavonic. The text may also include formulas and complex tables. Examples from the document collection are shown in Figure 1.

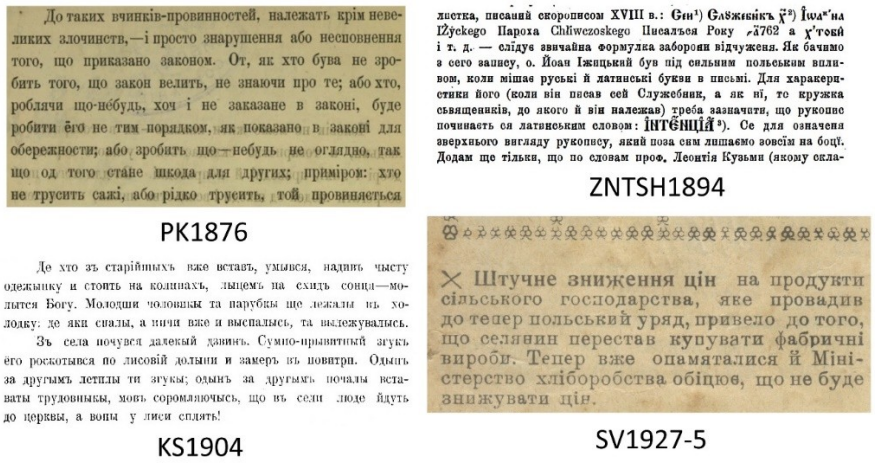


Figure 1: Examples of documents.

The sources used have their own graphical characteristics of text blocks. Binarized documents contain typical segmentation errors [22–27]. The documents ZNTSH1894 and PK1876 make extensive use of diacritical marks and different font types (Figure 2a, d). The black-and-white documents ZNTSH1894 and KS1904 contain broken letter strokes (Figure 2b). In contrast, document SV1927-5 shows character merging caused by ink spread (Figure 2c).

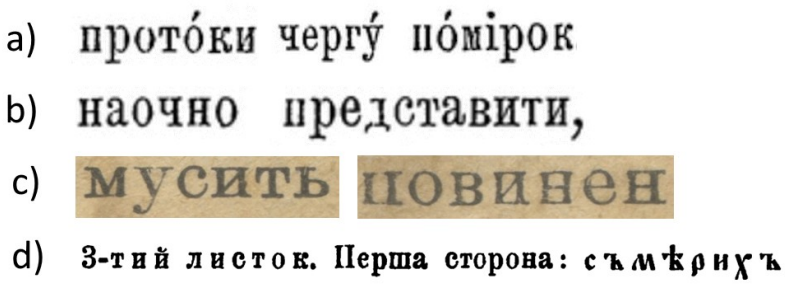


Figure 2: Graphical Features of the Documents.

The orthography used in the documents varies and differs from the standardized modern one in several ways. Document PK1876 contains Russian-language insertions within the Ukrainian text and uses “ъ” instead of the apostrophe, “и” instead of the modern “і” in native Ukrainian words, and “ё” instead of “ьо” or “йо”. In document ZNTSH1894, unlike modern orthography, the letter “і” is used instead of “и” in foreign words, while in native Ukrainian words “ї” is used instead of “і”; the hard “ґ” is also used systematically. This scholarly publication also contains quotations and insertions in Church Slavonic, Polish, and German. Document KS1904 uses “ы” instead of “и”, “и” instead of the modern “і”, the hard sign “ъ” at the ends of words, and the letter “ё” instead of “ьо”. The orthography of document SV1927-5 is the closest to the modern one, although Polish insertions also occur. This periodical has a complex multi-column layout, as well as decorative lines and separators directly within the text blocks.

The images in the dataset were manually segmented in order to exclude errors caused by scan defects, noise, loss of contrast, and skew. Drop caps, decorative lines, and column splitting were excluded. Historical documents were used for model testing and for the construction of training vocabularies. The training itself was performed on synthetic text lines generated on the basis of their orthographic and graphical characteristics.

4. Method

For text recognition, we used the free Tesseract software, version 5. It is characterized by reduced memory consumption, faster processing, and extended binarization capabilities. This is important for early printed books, since OCR quality in such materials depends not only on the model itself, but also on the system's ability to work correctly with uneven background, faded print, and non-uniform page contrast.

Tesseract 5 supports several scenarios for adapting a model to a new domain: fine-tuning based on an existing language model, cutting off the upper network layers followed by retraining, and full training from scratch. Fine-tuning can be effective even with a relatively small amount of new data if the target task is close to the source domain [28].

For Tesseract 5, training data are prepared at the level of text lines rather than individual characters [29]. This simplifies the creation of ground truth for historical materials. The main stages of training data preparation include text preparation, rendering new images or using existing ones, generating the unicharset file, creating the initial traineddata collection, generating model files, and training the network itself. One of the weak points of Tesseract remains its dependence on page segmentation quality.

The proposed recognition methodology evaluates the impact of document-specific features and model fine-tuning on OCR quality. The sequence of steps is shown in Figure 3.

At the first stage, the dataset is formed. The study uses pages that represent different old font styles, variants of historical orthography, different print quality, and typical digitization defects. The dataset includes both relatively clean pages and pages with problems such as text bleed-through from the reverse side of the sheet. Pages containing specific typographic elements, including drop caps, decorative initials, ornamental separators, and decorative rules, were annotated separately. For each image, ground truth is prepared, that is, a reference text that is later used for training and evaluation.

The second stage consists of image preprocessing. Its purpose is to reduce the influence of scan defects on subsequent recognition. Depending on the condition of the material, grayscale conversion, contrast normalization, binarization, page deskewing, margin cropping, and attempts to reduce the bleed-through effect were applied. Within the methodology, preprocessing was used as a technical procedure.

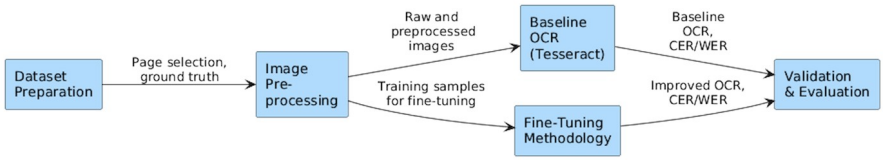


Figure 3: Document Recognition Methodology

The third stage involves the baseline recognition of documents using the standard Tesseract model without special fine-tuning. This step is necessary to establish the initial quality level against which all improved variants are later compared. If the model does not contain a complete target alphabet for recognizing new characters, it begins to produce errors:

1. replaces a historical character with the closest modern one,
2. omits it,
3. confuses it with another symbol.

The list of valid characters for a given recognition model is contained in the unicharset file. This is the registry of graphemes that the OCR model is able to process, including letters, digits, punctuation marks, and special symbols.

At this stage, the system is applied to the test pages. The OCR quality was evaluated using Character Error Rate (CER) and Word Error Rate (WER). The CER metric is calculated at the character level:

$$CER = \frac{S_c + D_c + I_c}{N_c}, \quad (1)$$

where:

S_c is the number of character substitutions,

D_c is the number of character deletions,

I_c is the number of character insertions,

N_c is the number of characters in the ground truth.

WER evaluates recognition errors at the word level:

$$WER = \frac{S_w + D_w + I_w}{N_w}, \quad (2)$$

where:

S_w is the number of word substitutions,

D_w is the number of word deletions,

I_w is the number of word insertions,

N_w is the number of words in the ground truth.

The obtained results show how well the standard OCR model performs under the conditions of old fonts, historical orthography, and defective scans.

The fourth stage consists in developing algorithms for fine-tuning Tesseract models. For this purpose, representative examples of text lines are prepared to cover the most problematic cases. These cases include letters from old typefaces and orthographic variants. During training, the basic grapheme registry is automatically expanded. Within the experiment, a modified version of the existing Tesseract model is created, after which OCR is performed on the validation and test synthetic subsets.

The fifth stage is validation and evaluation of the results. At this stage, two modes are compared: the standard Tesseract model on the original scans and the fine-tuned model on the original scans.

Historical book editions contain not only the main text but also a large number of specific typographic elements that significantly complicate optical character recognition. In such sources, alongside ordinary type, one may encounter a drop cap (initial), a decorative initial, an ornamental separator, and a decorative rule. These elements often become part of the text flow, causing the system to insert extra characters, distort paragraph boundaries, or disrupt the sequence of lines. As a result, recognition accuracy decreases, and further linguistic processing of the document becomes more difficult. To reduce recognition errors caused by decorative lines (Figure 4), a confidence-score analysis algorithm was applied at the preprocessing stage. For each image, the OCR system generates a TSV file containing detailed information about text location and confidence scores. Characters with a confidence score of zero are removed.

Table 2 presents the CER and WER values obtained when the documents were recognized using the standard Ukrainian language model. It should be noted that CER was calculated at the character level. Before calculating WER, the text was normalized: character case was ignored, word hyphenation was removed, and punctuation marks were not taken into account. This normalization makes it possible to focus the evaluation on lexical recognition errors while reducing the influence of formal differences in text formatting.



Figure 4: Removal of characters based on the confidence score

The original documents contain inserted characters that are not included in the standard model. Although the OCR system supports a mode that combines several base language models, this paper focuses specifically on fine-tuning a single base Ukrainian language model.

Table 2.
Recognition with the Standard UKR Language Model.

Document	CER, %	WER, %
PK1876	6,0	16,0
ZNTSH1894	6,9	21,1
KS1904	18,75	61,6
SV1927-5	3,5	18,9

5. Experiments

For training the Tesseract system, a Windows machine, Jupyter Notebook, Tesseract 5.5.0, Python 3.11.8, and the Windows Subsystem for Linux (WSL) environment were used. WSL was applied to ensure the correct use of Tesseract scripts and packages, which are primarily designed for Linux OS.

During fine-tuning, a new model may adapt well to a new font or a new character, but after a certain number of iterations, further training may begin to worsen the error on the base set [29]. Therefore, the Validation BCER parameter was used to select the best new model checkpoint during training. A checkpoint is an intermediate saved state of the model obtained after a certain number of training iterations. Since several checkpoints can be produced during fine-tuning, the final model was selected not by the last iteration, but by the lowest validation error. Validation BCER is a measure of character error on the validation subset. The validation subset consists of text lines that were not used directly to update the model weights during training. In the fine-tuning process, 90% of the prepared synthetic line images were used for model training, and 10% were used for validation.

Let M_t denote the model checkpoint obtained after iteration t, and let $BCER_{val}(M_t)$ be its character error rate on the validation subset. The selected model M^* was defined as:

$$M^* = \arg\min_{M_t} BCER_{val}(M_t), t \le T_{max}. \quad (3)$$

In our experiment, the upper bound for the fine-tuning duration was set to $T_{max} = 1500$ iterations. Thus, the final model was selected not only by the last training iteration, but by the minimum validation error observed among the generated checkpoints.

To fine-tune the model, an algorithm for generating synthetic text lines was developed (Figure 5). It combines three steps: targeted introduction of new characters, use of a word dictionary

extracted from the dataset documents, and addition of contrastive negative examples for graphically similar letters.

Let C_{new} be the set of new characters to be added to the base model, V_{new} the set of tokens containing at least one character from C_{new}, V_{conf} the set of contrastive tokens with graphically or positionally similar competing letters, and $V_{neu}$the set of neutral tokens. Each synthetic line is represented as a token sequence $L=(w_1, w_2, \ldots, w_n)$, where $n \in [2,8]$.

In the first step, characters from C_{new} are inserted into the training data either as isolated symbols or as short repeated sequences. This increases the frequency of newly introduced graphemes and ensures their stable inclusion in the generated unicharset file.

In the second step, the algorithm uses V_{new}, a dictionary of real words or plausible word forms extracted from the studied documents. In mixed synthetic lines, one token from V_{new} is used as an anchor, which guarantees the presence of at least one target grapheme. The remaining tokens are sampled from V_{new}, V_{neu}, or V_{conf}, after which the token order is randomized.

In the third step, the set V_{conf} is introduced to provide contrastive negative examples. It contains tokens without characters from C_{new}, but with graphically or positionally similar competing letters listed in Table 3. For example, for the character “ы”, this set included tokens with “ш”, “щ”, “ъ”, and related competing forms. These examples were used in both negative-only and mixed synthetic lines, allowing the model to learn the decision boundaries between the target graphemes and visually similar characters.

This generation scheme makes it possible to control the frequency of new graphemes, preserve lexical similarity to the original documents, and reduce substitution errors between newly introduced symbols and their closest visual competitors.

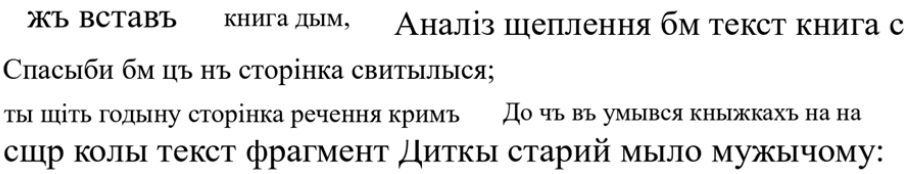


Figure 5: Synthetic text lines

To evaluate the proposed approach, experiments were conducted on a test set of document pages from our collection (Table 4). The proposed approach consists of two main components: (i) a synthetic text-line generation algorithm; and (ii) a fine-tuned Tesseract 5 Ukrainian OCR model adapted using these generated line images.

In the first experiment, the standard Tesseract Ukrainian language model was used as the baseline. In the second experiment, the fine-tuned model, modified by extending the character set and adapting it to historical orthography, was applied to the same original scans.

Table 3.
Examples of confusions for the standard model.

Ground truth	Confusion
ы	ъ ш щ
ъ	» ь Р Б
é	б
á	і

є	е
э	5 3 з

The evaluation showed that the standard ukr model provides acceptable results only for fragments in which the historical script is close to the modern one (Table 4). At the same time, words containing the characters listed in Table 3, as well as lines printed in old typefaces or affected by glyph degradation, produced a noticeable number of substitution, deletion, and insertion errors. In contrast, the proposed fine-tuned model reduced both character-level and word-level errors for all tested documents.

Table 4.
Comparison of the baseline model and the proposed fine-tuned model.

Language Model	KS1904	PK1876	ZNTSH1894	SV1927-5
Standard ukr, CER, %	18,75	6,0	6,9	3,5
Standard ukr, WER, %	61,6	16,0	21,1	18,9
Proposed Fine-tuned ukr, CER, %	**10,95**	5,0	5,9	3,0
Proposed Fine-tuned ukr, WER, %	**51,3**	12,0	11,1	13,9

For most documents, the improvement was moderate, within 1-2 percentage points for CER. The greatest effect was obtained for the most difficult document, KS1904. For this document, the baseline model produced the worst result, namely CER = 18.75% and WER = 61.6%. Applying the fine-tuned model in this worst-case scenario produced better results: CER = 10.95% and WER = 51.3%.

6. Conclusions

This study investigated the possibility of improving OCR quality for historical Ukrainian texts by fine-tuning the Tesseract 5 model on specially generated syn-thetic lines containing new characters. The experiment focused specifically on fine-tuning the OCR model without segmentation, block decomposition, or addi-tional image reconstruction stages. The experiments showed that the standard ukr model provides acceptable quality only for fragments whose script and orthogra-phy are close to the modern norm. In contrast, words containing the archaic characters listed in Table 3, as well as lines printed in old typefaces, showed a sub-stantial increase in substitution, deletion, and insertion errors.

Applying the fine-tuned model to the most difficult case improved the results to CER = 10.95% and WER = 51.3%. Thus, fine-tuning reduced the character error rate by 7.8 percentage points and the word-level error rate by 10.3 percent-age points. This corresponds to an improvement relative to the baseline model of approximately 41.6% in terms of CER and 16.7% in terms of WER. The obtained results confirm the value of adapting the model to historical script and outdated orthography.

In future work, the adapted OCR model should be combined with page seg-mentation and post-OCR correction.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] X. Wang, C. Reul, D. Christ, A. Hartelt, N. Balbach, M. Wehner, U. Springmann, C. Wick, C. Grundig, A. Büttner, F. Puppe, OCR4all—An Open-Source Tool Providing a (Semi-)Automatic OCR Workflow for Historical Printings, Applied Sciences 9 (2019). doi:10.3390/app9224853.

[2] K. Baierer, A. Büttner, E. Engl, L. Hinrichsen, C. Reul, OCR-D & OCR4all: Two Complementary Approaches for Improved OCR of Historical Sources, in: Y. Sumikawa, R. Ikejiri, A. Doucet, E. Pfanzelter, M. Hasanuzzaman, G. Dias, I. Milligan, A. Jatowt (Eds.), Proceedings of the 6th International Workshop on Computational History (HistoInformatics 2021) co-located with ACM/IEEE Joint Conference on Digital Libraries 2021 (JCDL 2021), CEUR-WS.org, 2021.

[3] K. Kettunen, M. Koistinen, J. Kervinen, Ground Truth OCR Sample Data of Finnish Historical Newspapers and Journals in Data Improvement Validation of a re-OCRing Process, LIBER Quarterly: The Journal of the Association of European Research Libraries 30 (2020) 1–20. doi:10.18352/lq.10322.

[4] A. Sangiacomo, H. Hogenbirk, R. Tanasescu, A. Karaisl, N. White, Reading in the mist: high-quality optical character recognition based on freely available early modern digitized books, Digital Scholarship in the Humanities 37 (2022) 1197–1209. doi:10.1093/llc/fqac014.

[5] K. Nikolaidou, M. Seuret, H. Mokayed, M. Liwicki, A Survey of Historical Document Image Datasets, arXiv (2022). URL: http://arxiv.org/abs/2203.08504. doi:10.48550/arXiv.2203.08504.

[6] D. Fleischhacker, R. Kern, W. Göderle, Enhancing OCR in historical documents with complex layouts through machine learning, International Journal of Digital Libraries 26 (2025) 3. doi:10.1007/s00799-025-00413-z.

[7] J. Martínek, L. Lenc, P. Král, Building an efficient OCR system for historical documents with little training data, Neural Computing and Applications 32 (2020) 17209–17227. doi:10.1007/s00521-020-04910-x.

[8] I. Gruber, L. Picek, M. Hlaváč, P. Neduchal, M. Hrúz, Improving Handwritten Cyrillic OCR by Font-Based Synthetic Text Generator, in: H. Moosaei, M. Hladík, P. M. Pardalos (Eds.), Dynamics of Information Systems, Springer Nature Switzerland, Cham, 2024, pp. 102–115. doi:10.1007/978-3-031-50320-7_8.

[9] T. Hegghammer, OCR with Tesseract, Amazon Textract, and Google Document AI: a benchmarking experiment, Journal of Computational Social Science 5 (2022) 861–882. doi:10.1007/s42001-021-00149-1.

[10] C. Vasantharajan, L. Tharmalingam, U. Thayasivam, Adapting the Tesseract Open-Source OCR Engine for Tamil and Sinhala Legacy Fonts and Creating a Parallel Corpus for Tamil-Sinhala-English, in: 2022 International Conference on Asian Language Processing (IALP), 2022, pp. 143–149. doi:10.1109/IALP57159.2022.9961304.

[11] A. Tikhonov, A. Rabus, Handwritten Text Recognition of Ukrainian Manuscripts in the 21st Century: Possibilities, Challenges, and the Future of the First Generic AI-based Model, KMHJ (2024) 226–247. doi:10.18523/2313-4895.11.2024.226-247.

[12] M. Bogdanović, M. F. Gligorijević, J. Kocić, L. Stoimenov, Improving Text Recognition Accuracy for Serbian Legal Documents Using BERT, Applied Sciences 15 (2025). doi:10.3390/app15020615.

[13] A. Beshirov, M. Dobreva, D. Dimitrov, M. Hardalov, I. Koychev, P. Nakov, Post-OCR text correction for Bulgarian historical documents, International Journal of Digital Libraries 26 (2025) 4. doi:10.1007/s00799-025-00415-x.

[14] O. B. Havrylyshyn, V. B. Repeta, I. Z. Myklushka, The analysis of the process of old books digitalization, Scientific Bulletin of UNFU 30 (2020) 106–110. doi:10.36930/40300318.

[15] N. Fischer, A. Hartelt, F. Puppe, Line-Level Layout Recognition of Historical Documents with Background Knowledge, Algorithms 16 (2023). doi:10.3390/a16030136.

[16] A. Spektor, Digitisation of genealogical documents based on automatic text recognition technology, Scientific Journal “Library Science Record Studies Informology” (2024) 56–68. doi:10.63009/lsrsi/4.2024.56.
[17] K. Tymoshenko, V. Vysotska, O. Kovtun, R. Holoshchuk, S. Holoshchuk, Real-Time Ukrainian Text Recognition and Voicing, in: COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems, April 22–23, 2021, Kharkiv, Ukraine, Kharkiv, Ukraine, 2021.
[18] K. Smelyakov, A. Chupryna, D. Darahan, S. Midina, Effectiveness of Modern Text Recognition Solutions and Tools for Common Data Sources, in: COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems, April 22–23, 2021, Kharkiv, Ukraine, Kharkiv, Ukraine, 2021.
[19] I. Ohiienko, Narysy z istorii ukrainskoi movy: systema ukrainskoho pravopysu = Outlines of the History of the Ukrainian Language, Volyn, Winnipeg, 1990.
[20] Yu. Tereshchenko, Dovhe XIX stolittia: sprotyv asymiliatsii, Tempora, Kyiv, 2022.
[21] P. Hrytsenko, Ukrainska mova v Rosii XIX – pochatku XX st.: shliakhy utverdzhennia, in: H. Boriak (Ed.), Ukrainska identychnist i movne pytannia v Rosiiskii imperii: sproba derzhavnoho rehuliuvannia (1847–1914). Zbirnyk dokumentiv i materialiv, Instytut istorii Ukrainy NAN Ukrainy, Kyiv, 2013, pp. XXXIX–LII.
[22] K. Ramalingam, R. Bhojan, Identification of Broken Characters in Degraded Documents, International Journal of Intelligent Engineering and Systems 11 (2018) 130–137. doi:10.22266/ijies2018.0630.14.
[23] Q. O. Mosa, M. F. Nasrudin, Broken character restoration using gradient vector flow and balloon force algorithm, Journal of Theoretical and Applied Information Technology 73 (2015).
[24] W. Peerawit, W. Yingsaeree, A. Kawtrakul, The utilization of closing algorithm and heuristic information for broken character segmentation, in: IEEE Conference on Cybernetics and Intelligent Systems, 2004, vol. 2, 2004, pp. 775–779. doi:10.1109/ICCIS.2004.1460686.
[25] D. R. R. Babu, M. Ravishankar, Manish Kumar, Aakash Raj, Kevin Wadera, Recognition of machine printed broken characters based on gradient patterns and its spatial relationship, in: 2010 3rd International Conference on Computer Science and Information Technology, 2010, pp. 673–676. doi:10.1109/ICCSIT.2010.5564682.
[26] K. Lipianina-Honcharenko, V. Yarych, A. Ivasechko, K. Yurkiv, T. Lebid, Evaluating the Effectiveness of Attention-Gated-CNN-BGRU Models for Historical Manuscript Recognition in Ukraine, in: The First International Workshop of Young Scientists on Artificial Intelligence for Sustainable Development, May 10–11, 2024, Ternopil, Ukraine, 2024, pp. 99–108.
[27] C. Clausner, A. Antonacopoulos, S. Pletschacher, Efficient and effective OCR engine training, International Journal on Document Analysis and Recognition (IJDAR) 23 (2020) 73–88. doi:10.1007/s10032-019-00347-8.
[28] How to train LSTM/neural net Tesseract, Tesseract documentation, 2026. URL: https://github.com/tesseract-ocr/tessdoc/blob/main/tess5/TrainingTesseract-5.md. Last accessed: 2026/05/25.