

Improving OCR for a Latvian Pronunciation Dictionary

Viesturs Jūlijs Lasmanis

University of Latvia, Institute of Mathematics and Computer Science

Raiņa bulvāris 29, Riga, Latvia

viesturs.lasmanis@lumii.lv

Abstract

This paper presents evaluation and data augmentation methods for improving OCR of the Latvian Language Spelling and Pronunciation Dictionary (LVPPV), a specialised dictionary containing numerous non-standard diacritic symbols used to transcribe Latvian pronunciation. To reduce the amount of manual correction required for OCR output, two new Tesseract v5 models were trained: one using a pre-existing annotated dataset and another using the same dataset supplemented with artificially generated training data. To enable more task-oriented model evaluation, two metrics were defined based on agreement between extracted lexeme–pronunciation pairs and two Latvian pronunciation resources: the Modern Latvian Language Dictionary and a rules-based phonetic transcriber. Both newly trained models outperform the previously published OCR model on these evaluation metrics, increasing transcription match rate from 71.81% to 87.56%. While synthetic data augmentation yields only limited improvements in overall transcription match rate, it substantially improves recognition of placenames containing uppercase letters, resulting in an 81% increase in correctly extracted placename pronunciations compared to the non-augmented model.

1 Introduction

The Latvian Language Spelling and Pronunciation Dictionary – “Latviešu valodas pareizrakstības un pareizrunas vārdnīca”, LVPPV (Ceplītis et al., 1995) is a published dictionary that contains approximately 80 000 lexemes with a detailed transcription of pronunciation data, including fully annotated intonations, which are not reflected in Latvian orthography (see Figure 1). This dictionary

also has a high-quality digital scan that can be used for information extraction using an OCR model.

dvest, dvešu, dves, dveš, dvešam, dvesāt, *pag.* dvesu, dvesām, dvesāt; *pa v.* dves, dvesiet; dvesdams [dvēzdāms], dvestu [dvēstu]
dvie/lis [dviēlis], -la
dvinga [dviŋga]
dvingains [dviŋgains]
dviņi [dviŋi], *usk.* dvī/nis, -ņa: s. -ne, -ņu
dviņkristāls [dviŋkristāls]
dviņledājs [dviŋledājs]
dviņogas [dviŋuôgas], *usk.* dviņoga
dviņsēkle/nis [dviŋsēklenis], -ņa
dviņser/de [dviŋsefde], -žu

Figure 1: Excerpt from the Latvian Language Spelling and Pronunciation Dictionary (LVPPV).

Although most of the dictionary is written in a typed font, it uses a varied list of combining diacritic marks not present in standard Latvian orthography, and there exist some handwritten corrections to the said diacritic marks, which provide an additional complication to the training of an accurate OCR model. The full list of additional symbols is depicted in Figure 2 with non-observed but theoretically possible symbols depicted in gray.

The pronunciation transcription used in the dictionary is the standard Latvian phonetic alphabet “Latviešu valodas fonētiskais alfabēts” (LFA) used in Latvian linguistic research, and defined within LVPPV (Ceplītis et al., 1995). However, this format is ill-suited for digital use due to the extensive use of diacritics, where many possible character-diacritic combinations do not result in a single Unicode character. To address this issue, a Latvian version of the SAMPA notation was introduced (Auziņa, 2005), which has since been adapted and used in the LVMLA (Latvian language machine-

à â ã ä ì î ï ï è é ê ë ï ü
 ù û ü ù ò ô õ ò ę ę ę ę ę
 m m r r : - * ·
 n n r r r
 n n i i
 ŋ ŋ ŋ i i

Figure 2: Non-standard characters used in LVPPV phonetic notation.

readable alphabet) notation. As shown in Table 1 all diacritic marks and symbols not present in the standard Latvian alphabet are replaced with symbols that can be easily typed and are not used elsewhere in the Latvian phonetic alphabet.

Lexeme (orthographic)	LFA	LVMLA	IPA
ezers	ezers	EzErs	æzærs
sāls	sāls	sāls	sɑ:ls
augš	aûks	auqks	auks
ola	uõla	uo=la	uolǎ

Table 1: Comparison of LFA, LVMLA, and IPA notations

Therefore, in order to effectively use the pronunciation data provided by the LVPPV dictionary, it is important not only to have an accurate OCR model for transcribing the digital copy of the dictionary, but also to have a method to extract the lexemes and convert phonetic transcriptions from LFA notation into LVMLA notation.

2 Related work

The article “OCR Challenges for a Latvian Pronunciation Dictionary” by [Strankale and Paikens \(2020\)](#) introduces an approach to solve the OCR problem of this exact dictionary by utilising version 4 of Tesseract – an open-source model, which they argue is well suited for this specific task.

However, due to a high error-rate present within this optically recognised dataset, the dataset has yet to be meaningfully utilised in linguistic or speech synthesis research. While the precise evaluation set WER is not stated within the article, the graph of the ‘verification set WER’ indicates it being slightly

above 10% ([Strankale and Paikens, 2020](#)), making it challenging and time-consuming to manually review. The authors identify diacritic recognition and character duplication as the primary sources of OCR errors.

Examining other articles related to diacritic recognition in non-standard language, Tesseract remains a reliable open-source choice. An article by [Ilić et al. \(2022\)](#) examines diacritic detection in Serbian dialectal dictionaries using Tesseract OCR, and an article by [Do \(2024\)](#) uses it for Vietnamese old text recognition.

[Ilić et al. \(2022\)](#) opt for a slightly different approach to diacritic detection in Serbian dialectal dictionaries – creating a pipeline that does text recognition and character bound box detection with Tesseract, but a separate CNN model that is responsible for diacritic detection. Although this approach is promising and has high accuracy for Serbian dictionary recognition, it is unclear how this pipeline-based approach compares in accuracy to the method of fine-tuning or retraining the top layers of the Tesseract model.

According to a study by [Hegghammer \(2022\)](#) for historical documents of English and Arabic Google DocumentAI and Amazon Textract models outperform fine-tuned Tesseract models. However, in the study by [Do \(2024\)](#) it was found that for dense book texts in Vietnamese, fine-tuned Tesseract OCR models outperform LLM based approaches such as GPT-4o, Azure DocAI and GoogleAI. The case of Vietnamese is also more relevant to the task at hand, because, while part of the standard orthography, the Vietnamese language utilises many diacritic combinations and is less resourced than English or Arabic texts. This does seem to suggest that fine-tuned Tesseract models are more suited for niche orthographies, such as the one used in LVPPV.

Existing synthetic data augmentation methods for OCR primarily focus on improving robustness across various different document collections by introducing variation in fonts, layouts, and imaging conditions ([Jaderberg et al., 2014](#); [Journet et al., 2017](#); [Belval, 2019](#)). Furthermore, many synthetic data generation strategies rely on post-OCR correction assisted by external vocabulary lists or language corpora, which relies on the existence of standard-language resources ([Guan and Greene, 2024](#)). In contrast, the OCR model developed in this study is optimized for the Latvian Lan-

guage Spelling and Pronunciation Dictionary, a single highly standardized resource with largely uniform typography and document structure. The core challenge therefore, is not the adaptation to various different documents, but instead the limited character combinations present within the available data. Therefore, the data augmentation strategy of image generation examined within this study primarily focuses on increasing the amount of under-represented character–diacritic combinations within the training dataset, rather than increasing robustness to changes in font or document style.

3 Methodology

Within the scope of this study, a new Tesseract v5 OCR model was trained by optimising the maximum iterations count. Two new models were trained using this principle – one using only the dataset used by [Strankale and Paikens \(2020\)](#), and another which was augmented with artificial data.

Models were trained using the Tesseract v5 training pipeline with the Latvian (lav) pretrained model as the starting model, network specification [Lfx256 O1c105], and a maximum of 300,000 training iterations. Training was performed on a workstation equipped with an NVIDIA GeForce RTX 4070 GPU (12 GB VRAM).

In order to compare the original [Strankale and Paikens \(2020\)](#) model results with the newly trained models, two different evaluation metrics based on available Latvian pronunciation resources were defined, implemented and utilised. The first compares phonetic transcriptions with those available in the MLVV dictionary ([Zuicena et al., 2025](#)), while the second performs a more limited comparison using a rules-based transcriber.

3.1 Datasets for model training

3.1.1 LVPPV dataset

The LVPPV dataset utilised in this article was created and manually annotated by [Strankale and Paikens \(2020\)](#). While the annotated dataset is not publicly available, the dataset and corresponding code-base was provided by the authors for further research. The full dataset consists of 42 box files, which includes 23 fully and 19 partially annotated pages.

The original annotation was created for training Tesseract version 4 model and thus contains box files denoting the pixel positions of individual char-

acters within the scanned page images. However, since Tesseract version 5 the training method of the OCR model has changed to require line-level inputs as opposed to character-level inputs ([Tesseract, 2025](#)).

This necessitated the transformation of the existing dataset by merging all characters that appear in the same row to create a text file containing the annotation for the respective line. After format conversion, the final processed dataset consisted of 3520 lines of text from the LVPPV dictionary.

3.1.2 Artificially augmented dataset

Analysis of the character–diacritic combinations present in the prepared dataset revealed that some combinations have low prevalence in the annotated LVPPV dataset as seen in Table 2. It is also important to note, that the list of characters depicted in the article by [Strankale and Paikens \(2020\)](#) is not exhaustive and there are some rare characters, which may appear in a select few lexemes such as the character ‘ŋ’ as shown in Figure 3.

Symbol	Count	Symbol	Count
ĩ	248	î	82
ĩ̃	9	ĩ̂	6
ř	91	ř̂	43
ĩ̃	68	ĩ̂	1
ĩ̃	188	ĩ̂	0
ŋ̃	3	ŋ̂	0
ĩ̃	15	ĩ̂	0

Table 2: Intonation marked consonant symbol frequencies in annotated LVPPV dataset.

vankšķēt [vaŋkšķēt], vankšķu, vankšķi, vankšķ, vankšķam, vankšķat, *pag.* vankšķēju, vankšķejām, vankšķējāt; *pav.* vankšķi, vankšķiet

Figure 3: Lexeme containing character-diacritic combination not present in annotated dataset.

Augmenting the dataset with artificially generated line images was explored as a means of increasing the representation of rare character–diacritic combinations in the training data. In an initial experiment, the generated strings consisted of random sequences of Latvian orthographic symbols followed by randomly generated LFA symbol sequences enclosed in square brackets, ensuring that both common and rare symbols were represented in the training set.

However, because Tesseract v5 is based on an LSTM architecture (Tesseract, 2025), it is able to utilise contextual information during recognition. Consequently, training on completely arbitrary symbol sequences may limit the usefulness of this contextual information. Therefore, a string generation method based on Latvian lexical forms was developed in order to produce more linguistically plausible character sequences.

Sequences were generated using a list of Latvian words from Tēzauris.lv online dictionary open data (Spektors et al., 2025) and transforming them by randomly appending intonation marking symbols in viable positions. These results were then structured as shown in Figure 4, which are theoretically possible sequences of LFA characters, although they do not correspond to the actual pronunciation of the respective word.

liekvārdība [liêkvārdîba]

Figure 4: Artificial line image generated for data augmentation.

Furthermore, upon manually examining the output of Strankale and Paikens (2020) model, it was observed that the model made mistakes when trying to recognise the uppercase letters of the orthographic spelling of placenames in pages 878–942 of the LVPPV. This is most likely due to removing the top layers of the OCR model for retraining purposes and the absence of uppercase letters within the annotated Strankale and Paikens (2020) dataset. Therefore, while conducting the artificial augmentation of the dataset it was decided to re-introduce a limited amount of uppercase letters by randomly capitalising some of the orthographic words in order to potentially resolve this issue.

These sequences were then passed to TextRecognitionDataGenerator (*trdg*) python library to be generated into PNG images (Belval, 2019). Since the original version of *trdg* does not support many of the necessary combining diacritic symbols a slightly modified version was used. Additionally, to ensure that the combining diacritics were properly rendered, Charis version 7.000 font¹ was utilised, as it is designed for broad multi-lingual use, has great support for combining diacritic marks and is a serif font similarly to the font used within the

LVPPV dictionary.

In order to test the potential of data augmentation, the aforementioned LVPPV dataset consisting of 3520 lines of annotated text was supplemented with 5000 artificially generated image files and their respective text files.

3.2 Evaluation metrics and data comparisons

Traditional OCR evaluation metrics such as Character Error Rate (CER) and Word Error Rate (WER) measure the fidelity of the reconstructed text. However, the intended downstream task of this work is extraction of lexeme–pronunciation pairs rather than reconstruction of dictionary pages. Character errors after the listed pronunciation may leave the pronunciation data usable, while an incorrect diacritic can invalidate the pronunciation despite having little impact on CER. Therefore, agreement between extracted lexeme–pronunciation pairs and external pronunciation resources provides a more task-relevant measure of OCR quality for the intended application.

3.2.1 Comparison with MLVV

Another pronunciation resource for the Latvian language is the MLVV dictionary, which is available digitally. However, this source is limited in the amount of lemmas with phonetic guides – 6117 as of Autumn 2025 version (Zuicena et al., 2025) – compared to the LVPPV dictionary. While Strankale and Paikens (2020) do mention the possibility of using the data from this dictionary for model evaluation in their article and the aforementioned code-base contains some code for realising this comparison, this is not explored within the published article.

The main advantage to this dataset is that the pronunciation information fully annotates intonation data and is stored within a structured XML file and can be easily accessed together with the corresponding lemma. This means that it is simple to extract a list of lexeme and phonetic transcription pairs, which can then be used to look for alignment with the OCR models predicted LVPPV text.

However, there are a few caveats to address regarding this approach. While the LFA phonetic notation is standardised in terms of character representation, the phonetic transcription used in MLVV is not fully phonetic as it does not represent consonant and consonant voicing assimilation, instead opting to preserve orthographic nuances. Table 3 illustrates some example words extracted from the

¹<https://software.sil.org/charis/release-7-000/>

LVPPV dictionary by the OCR model and manually checked, do not match the MLVV form due to differences in the usage of the LFA notation.

Orthography	LVPPV	MLVV
apdzīt	abdzīt	apdzīt
jokdaris	juōk*daris	juōkdaris
plecs	plēc:	plēcs
spožs	spuōš:	spuōžs

Table 3: Comparison of phonetic transcription differences between LVPPV and MLVV dictionaries.

This approach of comparing extracted OCR results to MLVV lexemes creates a non-standard metric, which while cannot determine the overall accuracy of the model, can be utilised effectively in order to compare different models in their effectiveness at extracting reliable lexeme-pronunciation pairs. This information arguably may be more useful in determining which model is better for lexeme-pronunciation pair extraction as CER may not be fully indicative of the OCR models effectiveness for the intended task.

3.2.2 Comparison with rules-based transcription

The comparison with MLVV lexemes is very useful in determining the models’ ability to correctly extract lexemes and their pronunciations with correctly determined intonation diacritic symbols. However, the MLVV set of lexemes with phonetic information is relatively small and cannot be used to examine a broader scope of lexemes.

In order to determine whether the trained models retain their accuracy on the entire LVPPV dictionary a comparison with rules-based generated phonetic transcriptions can be utilised. Latvian language pronunciation, except for a few nuances, is generally predictable from orthography and positional information, thus allowing rules-based transcriptions to be a viable comparison.

First, a list of Latvian language lexemes was extracted from the Tēzaur.lv online dictionary dataset (Spektors et al., 2025) and then further transformed using a Latvian rules-based G2P transcriber (UL IMCS, 2025), which applies positional rules present in the Latvian language to transcribe the pronunciation. It should be noted that this transcriber is limited in the ability to assess correct LVPPV transcriptions, as this evaluation cannot determine the correct intonation, pronunciations

of some vowels (/e/, /ē/, /o/) and if the word has non-standard stress.

By generating all possible vowel pronunciations and ignoring intonation and stress it is possible to compare whether the OCR recognised phonetic transcription aligns with the rules-based transcription on base character level. That is to say that, while these rules-based transcriptions cannot be used to evaluate models on their diacritic recognition accuracy, these transcriptions can therefore be used to evaluate the amount of other mistakes caused by such errors as – character doubling and other base character recognition mistakes.

This evaluation method can be of particular use in analysing the OCR model’s ability for transcribing pages 878–942 of the LVPPV, which contain placenames with uppercase letters. For the OCR data extracted by the model of Strankale and Paikens (2020) it can be observed that in most cases the uppercase letters are skipped or predicted as completely different characters, with most correct predictions happening for uppercase letters that are similar in shape to their lowercase counterparts. These two types of errors are ones that can be detected using only the rules-based approach, as this doesn’t involve diacritic detection.

4 OCR post-processing

In order to more accurately evaluate model performance, the LVPPV predictions and MLVV transcriptions were both post-processed normalising combining diacritic characters and applying Latvian consonant and consonant voicing assimilation rules where applicable. Diacritic character normalisation is absolutely necessary due to the combining diacritic characters potentially being recognised and applied in a different order compared to the MLVV or vice versa, resulting in the string being encoded in Unicode in differing ways, despite representing identical LFA symbols.

Since both LVPPV and MLVV dictionaries use the LFA transcription format with slight differences, the comparison was done in the LFA format. To ensure that the Strankale and Paikens (2020) model and the models trained within the scope of this article are comparable, each model’s OCR output files without post-processing were used as the input data and then post-processing rules were applied to all files to ensure that all post-processing and rules for lexeme-transcription pair extraction were identical. Identical methods were used since

various post-processing methods can significantly increase the amount of lexemes and transcriptions that are correctly recognised, and since it is known that the same dataset was used in model training this means that there isn't a risk in some post-processing rules applying to only a certain model.

Finally, in addition to the post-processing steps necessary for model evaluation, it is necessary to transform the Latvian phonetic alphabet (LFA) notation into LVMLA notation. This format is not only the desired storage format for the extracted data, but is also the format that is used in the rules-based phonetic transcriber. Therefore, a format converter is necessary in order to apply the rules-based metric described in Subsection 3.2.2.

For the purposes of transforming between these formats a Python program was written, which can in almost all cases correctly convert between the formats using simple rule-based transformations of the LFA transcription.

However, the inspection of the dictionary output revealed that a small number of entries require orthographic or morphological information for correct LVMLA conversion, particularly in compound words. Figure 5 shows one such example, where the expected vowel length cannot be determined from the LFA transcription alone.

Ziemeļīrija [ziēmēlīrija]

Figure 5: Lexeme where orthographic information is necessary to determine the correct LVMLA conversion.

5 Results and Evaluation

After the training process of the Tesseract v5 model, it is possible to run the in-built model evaluation function and get comparative CER and WER data (Tesseract, 2025). It is possible to observe from the plotted data (Figure 6) that the training and evaluation losses of this model at model checkpoints does not differ greatly from the model trained by Strankale and Paikens (2020) with the CER remaining at around 2%, while the WER at around 10%.

However, the models differ more substantially when evaluated using the task-specific metrics described in Subsection 3.2. Compared with the model of Strankale and Paikens (2020), the newly trained Tesseract v5 model increases the MLVV transcription match rate from 71.81% to 87.56% (Table 4) and the rules-based transcription match

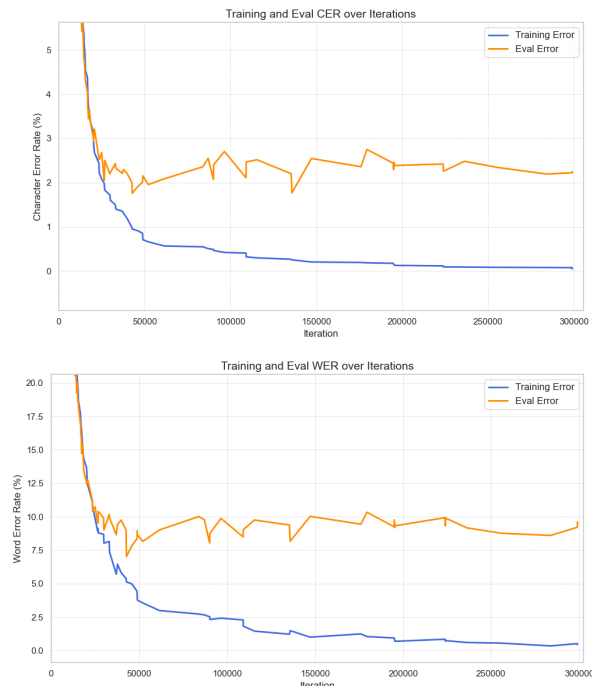


Figure 6: Tesseract v5 model checkpoint CER and WER for training and evaluation datasets depending on iteration count.

rate from 84.53% to 89.29% (Table 5).

Although the MLVV and rules-based evaluation metrics rely on different sources and therefore may have different potential causes of transcription mismatch, the most significant difference of the two methods is the fact that the MLVV metric includes diacritic information, while the rules-based method does not. The substantially larger improvement observed for the MLVV comparison (15.76 percentage points; Table 4) than for the rules-based comparison (4.76 percentage points; Table 5) therefore suggests that improved diacritic recognition contributes to the performance gain of the newly trained Tesseract v5 model.

Model	Lexeme & MLVV transcription match	Lexeme match	Transcription match rate
Strankale and Paikens (2020)	3571	4973	71.81%
Tesseract v5	4413	5040	87.56%
Tesseract v5 Artificially augmented	4356	5045	86.34%

Table 4: Comparison of MLVV lexeme transcription matches of trained OCR models.

The match rate metrics depicted in Tables 4, 5, 6 should not be interpreted as a standard OCR accuracy, since disagreements may arise from the LFA notation differences between LVPPV and MLVV or from other source data limitations rather than

Model	Lexeme & rule-based transcription match	Lexeme match	Transcription match rate
Strankale and Paikens (2020)	29815	35269	84.53%
Tesseract v5	31695	35498	89.29%
Tesseract v5 Artificially augmented	31783	35656	89.13%

Table 5: Comparison of rules-based transcription matches of trained OCR models.

recognition errors. Furthermore, for all of the models the rules-based extracted lexeme count depicted in Table 5 is significantly lower than that claim of 80 000 lexemes contained in the LVPPV (Ceplitis et al., 1995). However, it should be noted that many of the lexemes present in the LVPPV dictionary do not have a corresponding pronunciation guide, this being the case in situations when the pronunciation transcription in LFA is identical to the standard Latvian orthography.

For the most part the data augmented model achieves similar results to the base Tesseract v5 model. It achieves slightly lower agreement with the MLVV transcriptions (Table 4), while yielding a slightly larger number of matches according to the rules-based comparison (Table 5). Given the small magnitude of these differences and the fact that each model configuration was trained once, these results should be interpreted cautiously, as run-to-run variation was not evaluated. Therefore, the existing results do not provide evidence that synthetic augmentation improves overall transcription matching.

In contrast, the most notable improvement from artificially augmenting the dataset can be observed for placenames containing uppercase letters (Table 6). The number of matched placename transcriptions increased from 615 for the base Tesseract v5 model to 1,113 for the augmented model, with the augmented model also achieving a match rate closer to that observed in the main body of LVPPV. This substantial improvement suggests that the lack of uppercase characters in the original training data was a contributing factor to recognition errors in this section of the dictionary.

Model	Lexeme & rule-based transcription match	Lexeme match	Transcription match rate
Strankale and Paikens (2020)	412	524	78.63%
Tesseract v5	615	782	78.64%
Tesseract v5 Artificially augmented	1113	1294	86.01%

Table 6: Matches with rules-based transcriptions of placenames (LVPPV pages 878–942).

Overall, these results suggest that synthetic data

is most effective for addressing coverage deficiencies in the training set rather than uniformly improving recognition accuracy. While the augmentation substantially improved recognition of placenames containing uppercase letters (Table 6), it produced only a small increase in the absolute number of correctly matched transcriptions in the main body of the dictionary (Tables 4, 5), while the overall transcription match rate remained nearly unchanged. These findings suggest that synthetic augmentation is most beneficial when it increases coverage of under-represented patterns, but does not necessarily improve precision for patterns already represented in the training data.

5.1 Error Analysis

Manual inspection of the OCR output indicates that the most prevalent OCR errors reported by Strankale and Paikens (2020), namely character duplication and symbol confusion, remain present in the Tesseract v5 models. Table 7 illustrates examples of these and other recognition errors observed in the model output. The improved agreement with the rules-based transcriptions (Table 5) suggests a reduction in base-character-level transcription errors, a category that includes character duplication and base character recognition errors. As these individual error categories were not quantified separately, their relative frequency cannot be determined from the rules-based metric alone. Nevertheless, manual examination of the OCR output suggests that character duplication remains a recurring error, particularly for characters with non-standard diacritics marking intonation.

6 Conclusion

The precise accuracy of the models remains difficult to ascertain due to the target material being unique in character usage, and due to the annotated data being limited in scope. However, the alternate metrics defined and applied within this article can be successfully utilised to compare model performances. These metrics are important to successfully train new models which are capable of more accurately transcribing the Latvian Language Spelling and Pronunciation Dictionary, thereby reducing the amount of OCR output requiring manual correction.

The Tesseract v5 models trained within the scope of this article achieve better results under both metrics, suggesting improvements in both base-

Error type	Ground truth	OCR output
Character duplication	dzirks/ts [dzirksc]	dzirks/ts [dziirksc]
Diacritic recognition	vankšķēt [vaŋkšķēt]	vankšķēt [vaŋkšķēt]
Base character recognition	aplūkojums [aplūkuōjums]	aplūkojums [apūkuōjums]
Uppercase recognition	Adeleida [adele-ida]	mndeleida [adele-ida]
Bracket recognition	suverēns [suverēns]	suverēns [iverēns]

Table 7: Examples of the OCR error types observed in the non-augmented Tesseract v5 model output. Entries are shown in LFA notation and truncated to the segments relevant for lexeme–pronunciation pair extraction.

character-level and diacritic-sensitive recognition.

Despite the implementation of data augmentation in order to improve diacritic detection accuracy, the effects on overall model match rate from training Tesseract with the augmented dataset are minimal. However, the implemented data augmentation method does successfully improve the recognition of placenames containing uppercase letters within the dictionary, resulting in an 81% increase in correctly extracted placename pronunciations compared to the non-augmented model. Compared to the original OCR model of Strankale and Paikens (2020), the final augmented model more than doubles the number of correctly extracted placename pronunciations, highlighting the practical benefit of the improvements introduced in this work.

Although the methods developed in this study target LVPPV, the proposed task-oriented evaluation approach may also be applicable to other historical or linguistic resources where complete ground-truth transcriptions are unavailable but related lexical resources can provide partial external validation. Similarly, targeted synthetic augmentation may be useful for resources in which recognition difficulties arise from sparsely represented symbols or notation rather than document-level visual variation.

Further work should explore more effective data augmentation strategies and investigate whether varying the proportion of synthetic data in the training dataset yields better results. The robustness of the observed effects of data augmentation should also be evaluated across multiple training runs with different random seeds. Beyond model evaluation, future work should examine whether improvements in transcription matching translate into measurable reductions in manual post-editing effort.

References

Ilze Auziņa. 2005. *Latviešu valodas izrunas datormodelēšana*. Ph.D. thesis, University of Latvia.

Edouard Belval. 2019. Textrecognitiondata-generator. <https://github.com/Belval/TextRecognitionDataGenerator>.

L. Ceplītis, A. Miķelsons, T. Porīte, and S. Rāge. 1995. *Latviešu valodas pareizrakstības un pareizrūnas vārdnīca*. Avots.

Thao Do. 2024. Reference-based post-ocr processing with llm for diacritic languages. *arXiv e-prints*, pages arXiv–2410.

Shuhao Guan and Derek Greene. 2024. *Advancing post-OCR correction: A comparative study of synthetic data*. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6036–6047, Bangkok, Thailand. Association for Computational Linguistics.

Thomas Hegghammer. 2022. Ocr with tesseract, amazon textract, and google document ai: a benchmarking experiment. *Journal of Computational Social Science*, 5(1):861–882.

Velibor Ilić, Lenka Bajčetić, Snežana Petrović, and Ana Španović. 2022. Scydia-ocr for serbian cyrillic with diacritics. In *Dictionaries and Society. Proceedings of the XX EURALEX International Congress, 12-16 July 2022, Mannheim, Germany*, pages 387–400. Mannheim: IDS-Verlag.

Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*.

Nicholas Journet, Muriel Visani, Boris Mansencal, Kieu Van-Cuong, and Antoine Billy. 2017. *Doccreator: A new software for creating synthetic ground-truthed document images*. *Journal of Imaging*, 3:62.

Andrejs Spektors, Lauma Pretkalniņa, Normunds Grūzītis, Pēteris Paikens, Laura Rituma, Baiba Saulīte, Gunta Nešpore-Bērzkalne, Ilze Lokmane, Agute Klints, Madara Stāde, Mikus Grasmanis, Ilze Auziņa, Artūrs Znotiņš, Roberts Dargis, and Guntis Bārzdiņš. 2025. *Tēzauris.lv 2025 (autumn edition) (2025-09-22)*. CLARIN-LV digital library at IMCS, University of Latvia.

Laine Strankale and Pēteris Paikens. 2020. Ocr challenges for a latvian pronunciation dictionary. In *Human Language Technologies–The Baltic Perspective*, pages 199–206. IOS Press.

Tesseract. 2025. *Tesseract User Manual*. Available at <https://tesseract-ocr.github.io/tessdoc/>.

UL IMCS. 2025. [Parruna latvian g2p transcriber](#). Online pronunciation transcription tool for Latvian.

Ieva Zuicena, Ieva Auziņa, Santa Briede, Irēna Ilga Jansone, Ieva Kuplā, Gunta Lejniece, Ilga Migla, Laimdota Oldere, Ārija Ozola, Vija Požarnova, Sanda Rapa, Anitra Roze, Imants Šmidebergs, Dorisa Šnē, Māra Šnē, Agris Timuška, Mikus Grasmanis, Lauma Pretkalniņa, and Artūrs Znotiņš. 2025. Dictionary of contemporary latvian language (MLVV) (2025-06-21). CLARIN-LV digital library at IMCS, University of Latvia.