



# When Tesseract Brings Friends

## Layout Analysis, Language Identification, and Super-Resolution in the Optical Character Recognition of Medieval Texts

Vít Novotný<sup>1</sup> , Kristýna Seidlová<sup>2</sup>, Tereza Vrabcová<sup>1</sup>, and Aleš Horák<sup>1</sup> 

<sup>1</sup> Faculty of Informatics, Masaryk University  
Botanická 68a, 602 00 Brno, Czech Republic

{witiko,485431}@mail.muni.cz, hales@fi.muni.cz

<sup>2</sup> Department of Auxiliary Historical Sciences and Archive Studies  
Faculty of Arts, Masaryk University  
Arna Nováka 1, 602 00 Brno, Czech Republic

449852@mail.muni.cz

**Abstract.** In our previous article, we surveyed optical character recognition algorithms for medieval texts. However, accurate recognition remains an open challenge. In this work, we develop eight preprocessing techniques and we show that they improve ocr accuracy on medieval texts. We also produce and publish an open dataset of 51,351 scanned images and ocr texts with 120 human annotations for layout analysis and ocr evaluation, and 122 human annotations for language identification.

**Keywords:** Optical character recognition · Layout analysis · Language identification · Image super-resolution · Medieval texts

## 1 Introduction

The aim of the AHISTO project is to make documents from the Hussite era (1419–1436) available to the general public through a web-hosted searchable database. Although scanned images of letterpress reprints from the 19th and 20th century are available, accurate optical character recognition (ocr) algorithms are required to extract searchable text from the scanned images.

In our previous article [15], we have shown that the Tesseract 4 ocr algorithm was the second fastest and the most accurate among five different ocr algorithms. In this article, we investigate the impact of six preprocessing techniques on the accuracy of Tesseract 4. Additionally, we compare Tesseract 4 with three other ocr algorithms on the language identification task. Furthermore, we publish an open dataset [16] of scanned images and ocr texts with human annotations for layout analysis, ocr evaluation, and language identification.

In Section 2, we describe the related work in ocr preprocessing. In Section 3, we describe our three preprocessing techniques and our two evaluation tasks. In Section 4, we discuss the results of our evaluation. In Section 5, we offer concluding remarks and ideas for future work in the ocr of medieval texts.

## 2 Related Work

Today’s ocr algorithms use complex preprocessing pipelines that try to rid the scanned images of artefacts introduced by the printing process, the aging and

degradation of the paper, and the scanning process. In our work, we introduce eight additional preprocessing techniques based on layout analysis, language detection, and image super-resolution. In this section, we discuss the related work in each of these three areas.

## 2.1 Layout Analysis

In OCR preprocessing, layout analysis is one of the first steps, where the page is divided into areas of text and non-text. Two main types of methods exist:

1. *Bottom-up* methods either classify small patches of the scanned images and cluster patches of the same class into larger areas [26,17,7,3,4] or analyze whitespace to detect boundaries between areas [1,19,2]. They can adapt to non-rectangular areas but they often miss the global structure of the page.
2. *Top-down* methods [27,11,12] slice the page recursively into horizontal and vertical strips. They can discover large rectangular areas such as headings, columns, and paragraphs, but may fail to segment non-rectangular areas.

Tesseract 4 uses a hybrid technique [23] that first uses bottom-up techniques to detect the smaller areas in the page and then uses top-down techniques to group the smaller areas and decide their reading order.

## 2.2 Language Identification

In order to improve their accuracy, OCR algorithms need to identify the language of the text, so that they can use dictionaries and language models to narrow down the number of possible readings of the text.

Tesseract optimizes character segmentation and language modeling<sup>3</sup> [22,10]. The hypothesis with the highest combined score determines the language of a word. Older versions of Tesseract used separate models for character segmentation and language modeling and only combined their scores. Tesseract 4 uses a LSTM model that jointly optimizes both criteria.<sup>4</sup>

## 2.3 Image Super-Resolution

Traditionally, OCR engines used simple rule-based methods to maximize the signal-to-noise ratio in scanned images. Recent results show that image super-resolution techniques based on deep neural networks such as SRCNN [5] and the more advanced SRGAN [9] can be used as a preprocessing technique that improves OCR accuracy [8,13,24,14,6,20]. For more information about image super-resolution techniques, see another article from these proceedings on page 11.

<sup>3</sup> [https://tesseract-ocr.github.io/docs/das\\_tutorial2016/4CharSegmentation.pdf](https://tesseract-ocr.github.io/docs/das_tutorial2016/4CharSegmentation.pdf)

<sup>4</sup> [https://tesseract-ocr.github.io/docs/das\\_tutorial2016/7Building%20a%20Multi-Lingual%20OCR%20Engine.pdf](https://tesseract-ocr.github.io/docs/das_tutorial2016/7Building%20a%20Multi-Lingual%20OCR%20Engine.pdf)

### 3 Methods

In this section, we describe the ocr algorithms that we use in our experiments. We also describe our preprocessing techniques and how we evaluate them. Our experimental code is available online.<sup>5</sup>

#### 3.1 Optical Character Recognition

Besides Tesseract 4, we also use Tesseract 3, Tesseract 3 + 4, and Google Vision AI in our language identification experiments. We also use Google Vision AI in our image super-resolution experiments. For more information about the different ocr algorithms, see our previous article [15, Section 2].

#### 3.2 Scanned Image Dataset

In our previous article, we developed a dataset [15, Section 3.1] of 65,348 scanned image pairs in both low resolution (150 DPI) and high resolution (400 DPI).

To make it easy for others to reproduce and build upon our work, we use a subset of 51,351 scanned images (79%) from public-domain books in our experiments and we publicly release our dataset [16].

#### 3.3 Preprocessing

In this section, we describe our eight preprocessing techniques: two based on layout analysis, two based on layout identification techniques, and four based on image super-resolution.

*Layout Analysis* In our previous article, we showed that Google Vision AI [15, Section 4.2] is accurate but can fail to properly segment multi-column pages where Tesseract 4 does not.

We developed two layout analysis techniques based on *computational geometry* (see Algorithm 1) and *machine learning* (see Algorithm 2). We use our techniques to decide whether a page is single- or multi-column. Single-column pages are processed by Google Vision AI and multi-column pages by Tesseract 4.

<sup>5</sup> <http://gitlab.fi.muni.cz/xnovot32/ahisto-ocr>, file when-tesseract-brings-friends.ipynb

---

#### Algorithm 1: Layout analysis using computational geometry

---

**Result:** Whether the page contains a single column of text or multiple  
 Shoot seven horizontal rays in uniform vertical intervals over the page height;  
 Compute how many lines  $l_i$  in ocr output each ray  $i$  intersects;  
**if**  $\text{median}_{i \in \{2,3,\dots,6\}} l_i \leq 1$  **then**  
 | The page contains a single column of text;  
**else**  
 | The page contains multiple columns of text;  
**end**

---

---

**Algorithm 2:** Layout analysis using machine learning

---

**Result:** Whether the page contains a single column of text or multiple  
 Collect the  $x$ -coordinates of left and right boundaries of all lines in ocr output;  
 Combine the collected left and right boundaries into a set  $B$  of all boundaries;  
 Use `sklearn.svm.OneClassSVM` to remove outliers from  $B$ ;  
 Find the best number  $k \in \{0, 1, \dots, \min(10, |B|)\}$  of  $k$ -means clusters of  $B$  by  
 maximizing the Silhouette score;  
**if**  $k \leq 2$  **then**  
 | The page contains a single column of text;  
**else**  
 | The page contains multiple columns of text;  
**end**

---



---

**Algorithm 3:** Language identification based on paragraph languages

---

**Result:** Probability distribution  $\Pr(l)$  over the languages  $l$  of the page  
**foreach** candidate language  $l$  **do**  
 |  $\text{count}_l \leftarrow 0$ ;  
**end**  
**foreach** paragraph  $p$  with language  $l$  from the set of candidate languages **do**  
 |  $\text{count}_l \leftarrow \text{count}_l + \text{length of paragraph } p \text{ in characters}$ ;  
**end**  
**foreach** candidate language  $l$  **do**  
 |  $\Pr(l) \leftarrow \text{count}_l / \sum_{l'} \text{count}_{l'}$ ;  
**end**

---

*Language Identification* In 2006, Panák [18, Section 4.4] showed that using two-pass processing, where we first identify languages and then use the ocr algorithm with the identified languages can improve ocr accuracy. We developed two techniques for identifying page language using the languages of *paragraphs* (see Algorithm 3) and *words* (see Algorithm 4) in the ocr output of Tesseract 4.

In the first pass, we identified page languages using Tesseract 4 with two different sets of candidate languages based on the most frequent languages in our dataset: *three* (Czech, German, and Latin) and *nine* (Czech, German, Latin, Polish, French, English, Russian, Italian, and Slovak) candidate languages.

In the second pass, we use Tesseract 4 with languages  $l$  that were *detected* ( $\Pr(l) > 0\%$ ) and that satisfied  $\Pr(l) \geq t$  for a number of different thresholds  $t \in \{0\%, 25\%, 50\%, 75\%, 100\%\}$ . If none, then an empty ocr output is produced.

*Image Super-Resolution* The scanned images in the AHISTO project are often only available in the low resolution of 150 DPI. We use image super-resolution techniques to jointly upscale and reconstruct the images.

As our baseline preprocessing techniques, we use the original low-resolution and high-resolution images, and low-resolution images that were upscaled  $2\times$  using either bilinear interpolation or the Potrace vectorizer [21].

---

**Algorithm 4:** Language identification based on word languages

---

**Result:** Probability distribution  $\Pr(l)$  over the languages  $l$  of the page

```

foreach candidate language  $l$  do
  |  $\text{count}_l \leftarrow 0$ ;
end
foreach paragraph  $p$  with language  $l$  from the set of candidate languages do
  |  $\text{count}_l \leftarrow \text{count}_l + \text{length of paragraph } p \text{ in characters}$ ;
  | foreach word  $w \in p$  with language  $l'$  from the set of candidate languages do
  | |  $\text{count}_l \leftarrow \text{count}_l - \text{length of word } w \text{ in characters}$ ;
  | |  $\text{count}_{l'} \leftarrow \text{count}_{l'} + \text{length of word } w \text{ in characters}$ ;
  | end
end
foreach candidate language  $l$  do
  |  $\Pr(l) \leftarrow \text{count}_l / \sum_{l'} \text{count}_{l'}$ ;
end

```

---

As our actual preprocessing techniques, we use low-resolution images up-scaled either 2 $\times$  using SRCNN or 4 $\times$  using SRGAN. For SRCNN, we use two public SRCNN models<sup>6</sup> (further known as *Waifu2x*) that were pre-trained on drawn manga images with two different levels of noise removal: *low* (noise0) and *high* (noise3). For SRGAN, we use two models that we trained on the scanned images in our dataset and the born-digital PDF version of tome six of the book *Codex Diplomaticus et Epistolaris Regni Bohemiae* (further known as CDB VI) [25].

### 3.4 Evaluation

We evaluate our preprocessing techniques both intrinsically on the layout analysis and language detection tasks, and extrinsically on the OCR accuracy.

*Layout Analysis* For layout analysis, we report confusion matrices for the binary classification of pages as either single-column or multi-column. As our ground truth, we use 120 human-annotated pages that we publicly release in our dataset.

*Language Identification* For language identification, we report the percentage of pages (further known as *Accuracy@1*) where we correctly identified the primary language in the first pass. As our ground truth, we use 122 human-annotated pages<sup>7</sup> that we publicly release in our dataset.

*Optical Character Recognition* For OCR accuracy, we report the word error rate (further known as WER) [15, Section 3.2]. As our ground truth, we use 120 human-annotated pages that we publicly release in our dataset.

<sup>6</sup> <https://github.com/nagadomi/waifu2x/tree/master/models/cunet/art>

<sup>7</sup> <https://gitlab.fi.muni.cz/nlp/ahisto-language-detection>

## 4 Results

In this section, we report the results of our evaluation and we discuss the corpus of ocr texts that we created with our most successful preprocessing techniques,

### 4.1 Layout Analysis

Figure 1 shows that our simpler layout analysis technique that used computational geometry performed better on the intrinsic classification task and misclassified only two out of 120 (1.6%) pages. Our machine learning technique misclassified 31 out of 103 (30.1%) single-column pages as multi-column pages.

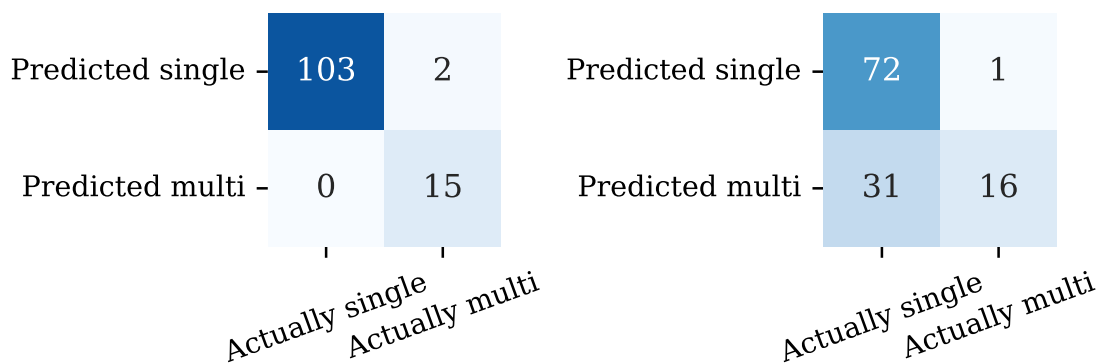


Fig. 1: Confusion matrices of computational geometry (left) and machine learning (right) layout analysis techniques

Figure 2 confirms our observation that although Google Vision AI performs generally worse than Tesseract 4, it performs significantly better on single-column pages and fails catastrophically on multi-column pages. By combining Google Vision AI and Tesseract 4 with our layout analysis technique using computational geometry, we receive significant improvements to the ocr accuracy.

### 4.2 Language Identification

Figure 3 shows that Google Vision AI performs significantly better than Tesseract on the intrinsic page language identification task. For Tesseract, using nine candidate languages with the *word* language identification technique consistently outperformed other configurations.

Figure 4 shows that using two-pass processing with nine candidate languages, the *paragraph* language identification technique that limits the number of detected languages, and the 0% threshold that only removes candidate languages that weren't at all detected can improve the ocr accuracy of Tesseract 4.

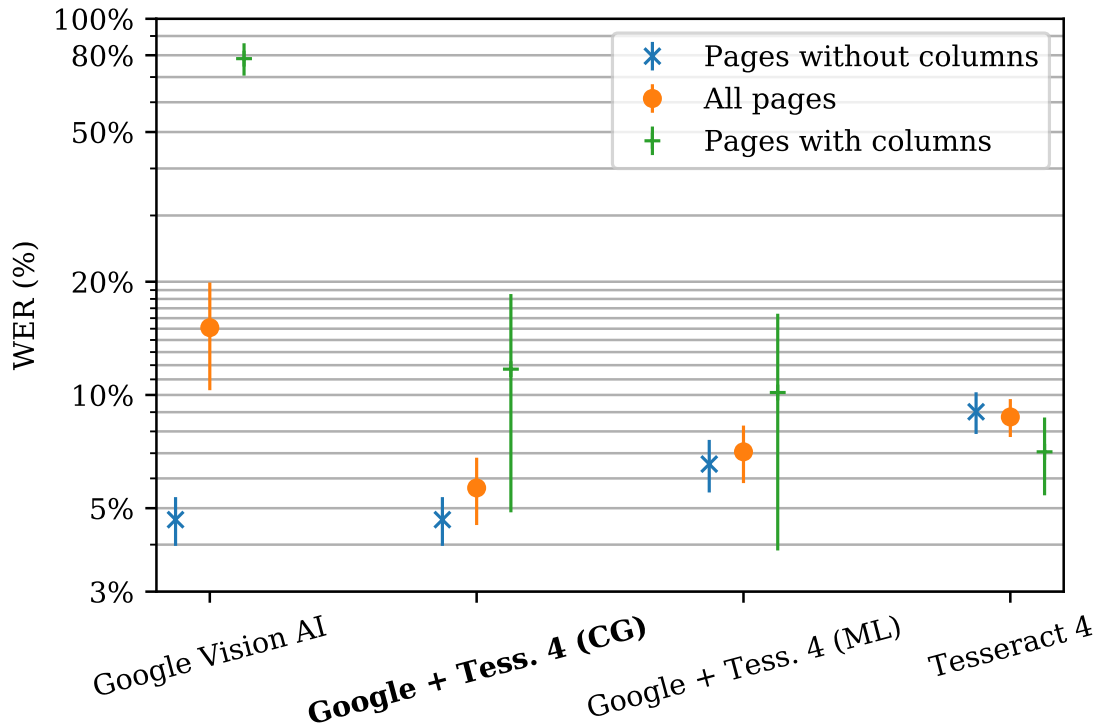


Fig. 2: ocr accuracies of Google Vision AI and Tesseract 4 alone and combined using two different layout analysis techniques (computational geometry and machine learning) on different subsets of pages. The best technique is **bold**.

### 4.3 Super-Resolution

Figure 5 shows that Google Vision AI does not particularly benefit from image super-resolution techniques. In contrast, Tesseract 4 always achieves better ocr accuracy with super-resolution techniques than with low-resolution images and outperforms even high-resolution images with the Waifu2x and SRGAN image super-resolution techniques. The pre-trained Waifu2x models outperform our SRGAN models, which may indicate a lack of training data.

### 4.4 Text Corpus

We combined our most successful preprocessing techniques: layout detection using computational geometry, two-pass processing with 0% threshold, nine candidate languages, and *paragraph* language identification technique, and the Waifu2x image super-resolution technique with high noise removal.

With the combined techniques, we achieved 5.42% WER compared to 8.74% with no preprocessing. Additionally, we also produced 51,351 ocr texts that we include in our dataset.

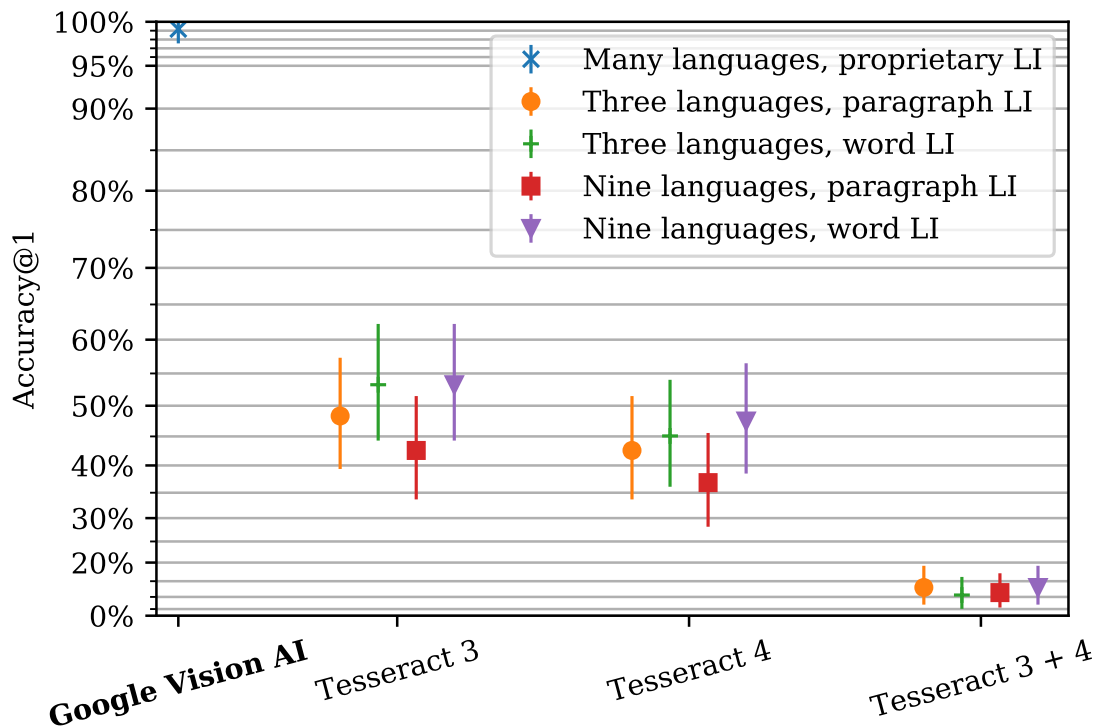


Fig. 3: Language identification accuracies of four different ocr engines using two different sets of candidate languages (three and nine) and two different language identification techniques (paragraph and word). The best ocr engine is **bold**.

## 5 Conclusion and Future Work

The ocr of scanned images for contemporary printed texts is widely considered a solved problem. However, the ocr of early printed books and reprints of medieval texts remains an open challenge. In our work, we developed eight preprocessing techniques in three different areas and we showed that they can improve the ocr accuracy on medieval texts. We also published an open dataset [16] of 51,351 scanned images and ocr texts with 120 annotations for layout analysis and ocr evaluation and 122 annotations for language identification.

In our work, we only used language identification preprocessing techniques based on language identification for individual pages. However, in printed collections of multilingual texts, ocr accuracy may be improved by processing smaller areas of the page separately. Additionally, we would produce an empty ocr output when no languages were detected or passed the confidence threshold, just disabling the language models in Tesseract may give better results.

**Acknowledgements.** This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2018101 and by TAČR Éta, project number TL03000365. The first author’s work was also funded



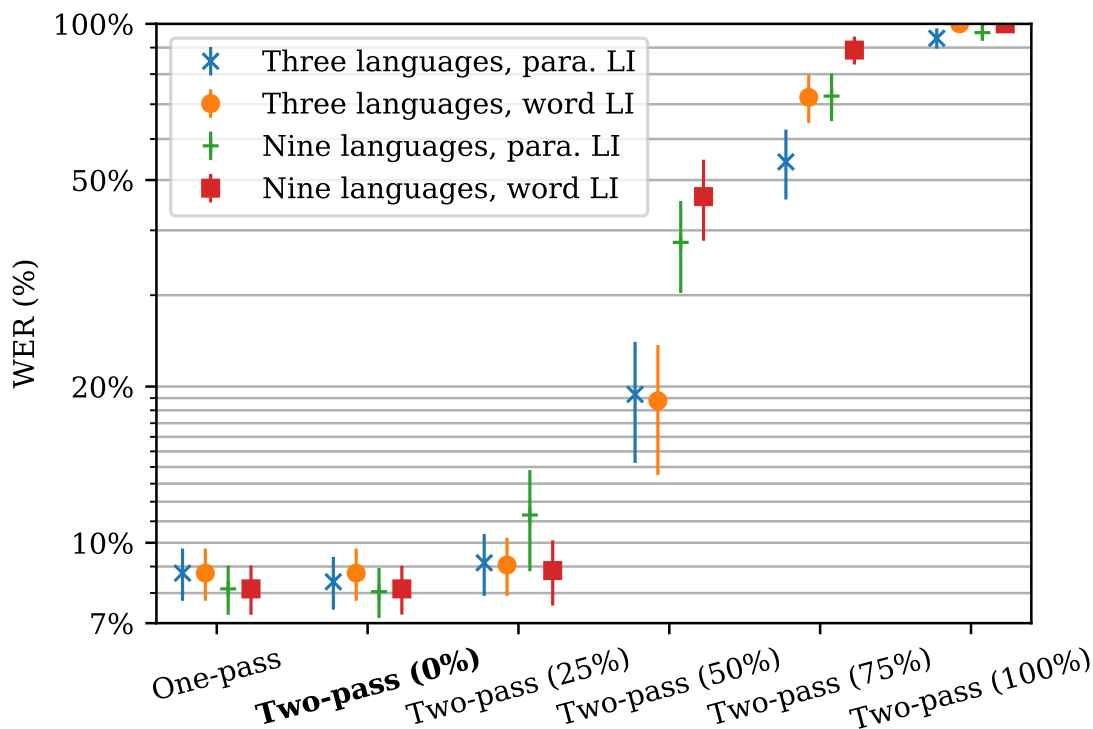


Fig. 4: ocr accuracies of Tesseract 4 using two different sets of candidate languages (three and nine) and two different page language identification techniques (paragraph and word). The best technique is **bold**.

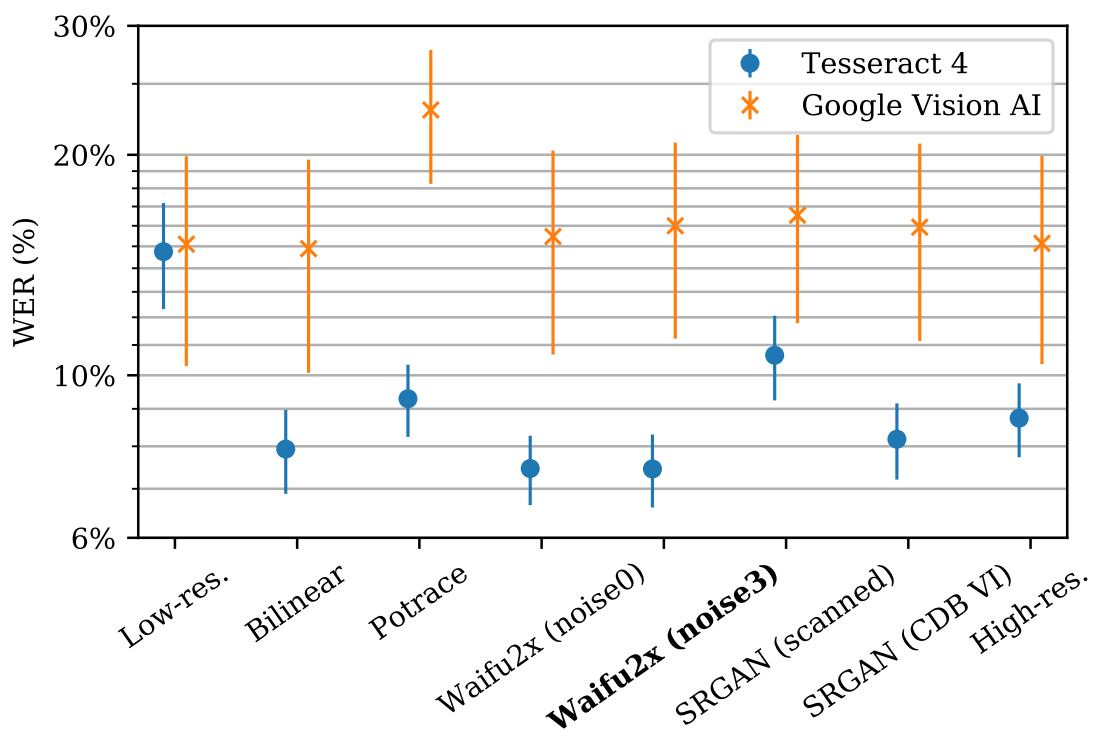


Fig. 5: ocr accuracies of Google Vision AI and Tesseract 4 using four different baselines and four different image super-resolution techniques. The best technique is **bold**.

by the South Moravian Centre for International Mobility as a part of the Brno Ph.D. Talent project.

## References

1. Baird, H.S., Jones, S.E., Fortune, S.J.: Image segmentation by shape-directed covers. In: ICPR. vol. 1, pp. 820–825. IEEE (1990)
2. Breuel, T.M.: Two geometric algorithms for layout analysis. In: Int. workshop on document analysis systems. pp. 188–199. Springer (2002)
3. Chen, M., Ding, X., Wu, Y.: Unified HMM-based layout analysis framework and algorithm. *Science in China, Series F: Information Sciences* **46**(6), 401–408 (2003)
4. Chowdhury, S., Mandal, S., Das, A., Chanda, B.: Segmentation of text and graphics from document images. In: ICDAR. vol. 2, pp. 619–623. IEEE (2007)
5. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *Computer Vision – ECCV 2014*. pp. 184–199 (2014)
6. Fu, Z., Kong, Y., Zheng, Y., Ye, H., Hu, W., Yang, J., He, L.: Cascaded detail-preserving networks for super-resolution of document images. In: ICDAR. pp. 240–245. IEEE Computer Society (2019)
7. Kise, K., Sato, A., Iwata, M.: Segmentation of page images using the area Voronoi diagram. *Computer Vision and Image Understanding* **70**(3), 370–382 (1998)
8. Lat, A., Jawahar, C.V.: Enhancing OCR accuracy with super resolution. In: ICPR. pp. 3162–3167. IEEE (2018)
9. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proc. of the IEEE conf. on computer vision and pattern recognition. pp. 4681–4690 (2017)
10. Lee, D.S., Smith, R.: Improving book OCR by adaptive language and image models. In: Int. Workshop on Document Analysis Systems. pp. 115–119. IEEE (2012)
11. Nagy, G., Seth, S.C.: Hierarchical representation of optically scanned documents. In: ICPR. pp. 347–349 (1984)
12. Nagy, G., Seth, S., Viswanathan, M.: A prototype document image analysis system for technical journals. *Computer* **25**(7), 10–22 (1992)
13. Nakao, R., Iwana, B.K., Uchida, S.: Selective super-resolution for scene text images. In: ICDAR. pp. 401–406 (2019)
14. Nguyen, K.C., Nguyen, C.T., et al.: A character attention generative adversarial network for degraded historical document restoration. In: ICDAR. pp. 420–425 (2019)
15. Novotný, V.: When tesseract does it alone: Optical character recognition of medieval texts. In: Horák, A., Rychlý, P., Rambousek, A. (eds.) *Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2020*. pp. 3–12. Tribun EU (2020)
16. Novotný, V., Seidlová, K., Vrabcová, T., Horák, A.: A human-annotated dataset of scanned images and OCR texts from medieval documents (2021), <https://nlp.fi.muni.cz/projects/ahisto/ocr-dataset>, [cited 2021-11-16]
17. O’Gorman, L.: The document spectrum for page layout analysis. *IEEE Transactions on pattern analysis and machine intelligence* **15**(11), 1162–1173 (1993)
18. Panák, R.: Digitalizace matematických textů. Master’s thesis, Faculty of Informatics, Masaryk University (2006), <https://is.muni.cz/th/pspz5/>
19. Pavlidis, T., Zhou, J.: Page segmentation and classification. *CVGIP: Graphical models and image proc.* **54**(6), 484–496 (1992)

20. Ray, A., Sharma, M., et al.: An end-to-end trainable framework for joint optimization of document enhancement and recognition. In: ICDAR. pp. 59–64 (2019)
21. Selinger, P.: Potrace: a polygon-based tracing algorithm (2003), <http://potrace.sourceforge.net/potrace.pdf>, [cited 2021-11-07]
22. Smith: An overview of the tesseract OCR engine. In: ICDAR. pp. 629–633. IEEE (2007)
23. Smith: Hybrid page layout analysis via tab-stop detection. In: ICDAR. pp. 241–245. IEEE (2009)
24. Su, X., Xu, H., Kang, Y., Hao, X., Gao, G., Zhang, Y.: Improving text image resolution using a deep generative adversarial network for optical character recognition. In: ICDAR. pp. 1193–1199 (2019)
25. Sviták, Z., Krmíčková, H., Krejčíková, J., Friedrich, G.: *Codex diplomaticus et epistolaris Regni Bohemiae. Tomi VI.* Academia (2006)
26. Wahl, F.M., Wong, K.Y., Casey, R.G.: Block segmentation and text extraction in mixed text/image documents. *Computer graphics and image proc.* **20**(4), 375–390 (1982)
27. Wong, K.Y., Casey, R.G., Wahl, F.M.: Document analysis system. *IBM journal of research and development* **26**(6), 647–656 (1982)