Improving Optical Character Recognition of Finnish Historical Newspapers with a Combination of Fraktur & Antiqua Models and Image Preprocessing

Mika Koistinen
National Library of Finland
The Centre for Preservation and Digitisation
mika.koistinen@helsinki.fi

Kimmo Kettunen
National Library of Finland
The Centre for Preservation and Digitisation
kimmo.kettunen@helsinki.fi

Tuula Pääkkönen
National Library of Finland
The Centre for Preservation and Digitisation
tuula.pääkkönen@helsinki.fi

Abstract

In this paper we describe a method for improving the optical character recognition (OCR) toolkit Tesseract for Finnish historical documents. First we create a model for Finnish Fraktur fonts. Second we test Tesseract with the created Fraktur model and Antiqua model on single images and combinations of images with different image preprocessing methods. Against commercial ABBYY FineReader toolkit our method achieves 27.48% (FineReader 7 or 8) and 9.16% (FineReader 11) improvement on word level.

Keywords: Optical Character Recognition, OCR Quality, Digital Image Processing, Binarization, Noise Removal, Tesseract, Finnish, Historical Documents

1 Introduction

These days newspapers are published in digital format (born digital). However, historical documents were born before the digital age and need to be scanned first and then extracted as text from the images by using optical character recognition (OCR). Currently the National Library of Finland (NLF) has over 10 million scanned historical newspaper and journal pages and there is need to improve the OCR quality, because it affects the usability and information retrieval accuracy for individual users, researchers and companies at the Digi newspaper collection.[1]

NLF's current document collection is discussed more in detail in Kettunen et al. (2016). Usually OCR quality of a historical document collection is many times on the level of 70-80% word accuracy. Tanner et al. (2009) estimated that OCR quality of the British 19th century newspaper collection has 78% word accuracy. According to Järvelin et al. (2015) improved OCR quality would improve the usability and information retrieval results for the Digi collection. Evershed and Fitch (2014) show that over 10% point OCR word accuracy improvement could improve the search recall by over 9% point on OCRed historical English documents. According to Lopresti (2009) better OCR quality would improve the possiblities to utilize other text analysis methods for example sentence boundary detection, tokenization, and part-of-speech tagging. Also other methods such as named entity recognition, topic modeling and machine translation could benefit from better quality input data.

There are many existing commercial and open source OCR tools available. Open source tools are attractive choices, because they are free and open for improvement. For example Tesseract[2] is an open source OCR engine, that is combined with Leptonica image library processing. It has models for over 100 languages. Some of the open source OCR tools e.g. Tesseract, Cuneiform, and OCRopus are discussed more in detail in (Smitha et al., 2016; Smith, 2007). From commercial tools ABBYY FineReader is one of the most known ones.

In our work, we develop a Tesseract Finnish Fraktur model using an existing German Fraktur model[3] as a starting point. We compare the resulting model with the commercial ABBYY FineReader toolkit. Previously, OCR quality of Tesseract and ABBYY FineReader has been compared by Heliński et al. (2012) on Polish historical documents. In their experiments, Tesseract outperformed FineReader on good quality pages containing Fraktur fonts, while FineReader performed better on Antiqua fonts and bad quality pages.

In addition to developing a new Finnish Fraktur model we study the effect of various preprocess-

[1] digi.kansalliskirjasto.fi

[2] https://code.google.com/p/tesseract-ocr/

[3] https://github.com/paalberti/tesseract-dan-fraktur

ing methods employed to improve the image quality on final OCR accuracy. Previously these kind of methods have shown to yield improvements in the image and/or OCR quality in several works (Smitha et al., 2016; Ganchimeg, 2015; El Harraj and Raissouni, 2015; Wolf et al., 2002; Sauvola and Pietikäinen, 1999).

The rest of the paper is organized as follows. In Section 2, we discuss challenges and methods related to scanned newspaper image quality improvement and Tesseract model teaching. The developed method and its evaluation are then discussed in Sections 3 and 4, respectively. Finally, we present conclusion on the work in section 5.

2 Challenges

OCRing of Finnish historical documents is difficult mainly because of the varying quality newspaper images and lack of model for Finnish Fraktur. Also the speed of the OCR algorithm is important, when there is need for OCRing a collection containing millions of documents. Scanned historical document images have noise such as scratches, tears, ink spreading, low contrast, low brightness, and skewing etc. Smitha et al. (2016) present that document image quality can be improved by binarization, noise removal, deskewing, and foreground detection. Image processing methods are briefly explained next as a background for improving the scanned image quality.

2.1 Improving the image quality by image processing methods

Digital images can be processed either by sliding a rectangular window through image to modify it's pixel values inside the window (local) or the whole image can be processed at once (global).

Binarization is image processing method that turns grayscale image pixels into binary image. The pixels in image are transferred to either black (0) or white (255) by a threshold value. Binarization methods according to Segzin and Sankur (2004) are based on histogram shape-based methods, clustering-based methods such as Otsu (1979), entropy-based, object atribute-based, spatial methods and local methods such as Niblack (1986); Sauvola and Pietikäinen (1999). Tesseract toolkit uses Otsu's algorithm for binarization as a default, which is not always the best method for degraded documents.

Wolf et al. (2002) developed image binarization algorithm with much better recall and slightly better precision. The method is based on Niblack and Sauvola algorithms. Niblack's algorithm is using rectangular window that is slided through the image. It's center pixel threshold (T) is calculated by using the mean (m) and variance (s) of the values inside the window.

$$T = m + k * s, \tag{1}$$

where k is constant set to 0.2. This method unfortunately usually creates noise to areas where there is no text. To avoid noise Sauvola's method included hypothesis on gray values of text and background pixels, thus modifying the formula into

$$T = m * (1 - k * (1 - \frac{s}{R})), \tag{2}$$

where R is the dynamics of standard deviation that is set to 128. However this hypothesis does not hold in every image documents such as variable contrast degraded documents. Therefore the formula was changed to normalize the contrast and mean gray level of the image.

$$T = m - k * (1 - \frac{s}{R}) * (m - M)), \tag{3}$$

where R is the maximum standard deviation from all of the windows. M is the minimum graylevel of the image.

Smoothing or blurring is a method to attenuate the most frequent image pixels. It is typically used to reduce image noise. Gonzales and Woods (2002) (p. 119-125) presents smoothing windows (Figure 1),

$$\frac{1}{9} * \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} * \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Figure 1: Smoothing windows

where the former is the average of gray levels and latter is the weighted average approach (window is rough approximation of a Gaussian function). These windows can be slided through image and the output is smoothed image. Ganchimeg (2015) presents history of document preprocessing methods noise removal and binarization. These methods can be based on thresholding, fuzzy methods, histograms, morphology, genetic algorithms, and marginal noise removal. The filtering techniques Gaussian, Sharpen, and Mean are compared, and it is noted that Gaussian blur is best for creating clear and smooth images. Droettboom (2003) mentions problem with broken and touching characters in recognizing older documents, and proposes broken character connection algorithm using k-nearest neighbours, which is able to find and join 91% of the broken characters. Original, blurred and some binarization methods for four different images can be seen below in Figure 2.

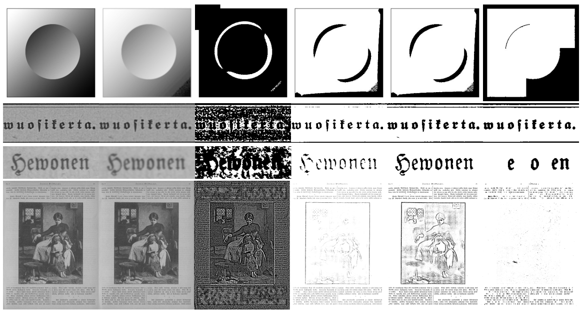

Figure 2: a) Grayscale(left) , b) Gaussian blur c) Niblack, d) Sauvola e) WolfJolion, f) Howe 3[6][7].

Image histogram presents image grayscale values(x) between 0-255 and frequencies(y) for all of the image pixels, where value 0 is black and 255 is white. It is a common way to visualize the image contents. According to Krutsch and Tenorio (2011) histogram equalization methods are Histogram expansion, Local area histogram equalization, Cumulative Histogram Equalization, Par Sectioning and Odd Sectioning. These methods try to improve the dynamic range of the image. High dynamic range means same as high contrast. Typically images with low contrast are visually very dull, grayish.

Linear normalization is also called contrast stretching or histogram expansion. Below in Equation 4, a linear transformation function is shown. This function is utilized to map the original images pixel values for broader range.

$$y = y_{max} * \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4}$$

Histogram equalization is another more advanced method for enchancing image contrast. It differs from linear transformation by using statistical probability distributions instead of linear functions. Image pixel probability density function (PDF) and cumulative density function (CDF) are utilized to to calculate new image gray levels. However, global methods had problems with varying intensities inside the image. Thus an Adaptive Histogram Equalization (AHE) was developed. It partitions the image usually to 8 x 8 windows. For each of these image windows sub-histogram is used to equalize each window separately, but still AHE created problematic artifacts inside regions that contained noise, but no text.

Pizer et al. (1990) developed Contrast Limited Adaptive Histogram Equalization (CLAHE) that was limiting the error in the AHE. CLAHE uses the slope of transform function. It clips the top part of histogram (by predefined clip limit value) before calculating the CDF for each of the images in sliding window (see figure 3). The clipping reduces the over-amplification AHE had. Result image depends from window size and clip limit, and are selected by the user.

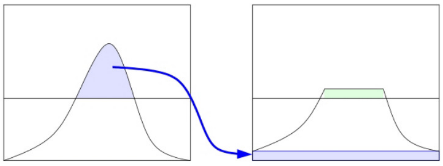

Figure 3: CLAHE histogram clipping from Stanhope (2016)

Below in Figure 4 results of contrast improvement methods are shown. CLAHE histogram has clearly equalized more than the linear normalized image.

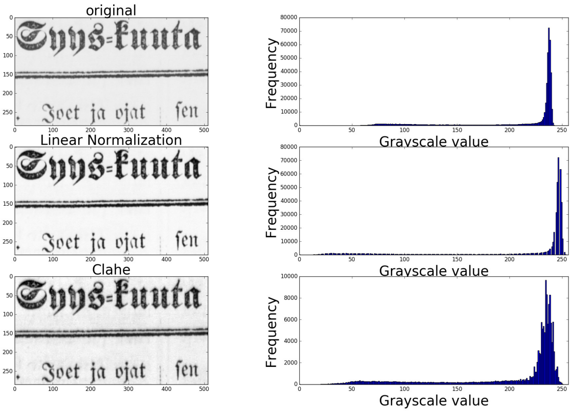

Figure 4: Original, Linear Normalized and Contrast Limited Adaptive Histogram Equalized image and their histograms

Skewing can also be a problem and it can be solved by skew detection and deskewing. According to Parashar and Sogi (2012) common methods for skew detection are Hough transform, projection profile and principal component analysis. Skew can be single or multiple. In multiple skew multiple parts of the document are skewed in different angle, and in single skew the image has been skewed only to one angle. We are dealing with multiple skew documents. More information about deskewing can be found from (Smith, 1995; Makkar and Singh, 2012). Our method does not use deskewing so far, but it could be included to possibly improve the accuracy.

Multiple other image processing methods exist, too. Image processing methods are widely researched for the use of historical document processing. For example annual International Conference on Document Analysis and Recognition DIBCO competition offers new methods for historical document image processing (Pratikakis et al., 2013; Ntirogiannis et al., 2014; Howe, 2013).

3 Method

We created a model for Finnish Fraktur character recognition for Tesseract. After that we run Tesseract OCR with different image preprocessing methods and chose the best one, by comparing the average confidence measure documents have. We used the hOCR-format[8], which is an open standard for presenting OCR results and it has confidence value for each word given by the used OCR tool. The average of Tesseract's word confidences in a document is used in this method.

Creation of Finnish Fraktur model was started by using German Fraktur model as a baseline. The Fraktur model was iteratively improved. The characters that had most errors were improved in training data boxes (letters and two letter combinations). Then Tesseract is run 1 to N times with the developed Finnish Fraktur model and already existing Finnish Antiqua model[9] in dual model mode, where it selects best choice from Fraktur and Antiqua results.

The images passed into Tesseract OCR on each run are either no processing, or processed by some image processing methods. In case of 1 run that run is selected as a final document, and in case of 2-N runs, the final resulting document is selected by the highest document confidence measure value. Our proposed solution for preprocessing and OCRing using the obtained Finnish Fraktur model can be seen below in Figure 5. Tesseract has build in Otsu's binarization so actually no processing means running the Otsu's algorithm before OCR. Otsu is also run inside Tesseract before OCR for all other methods.

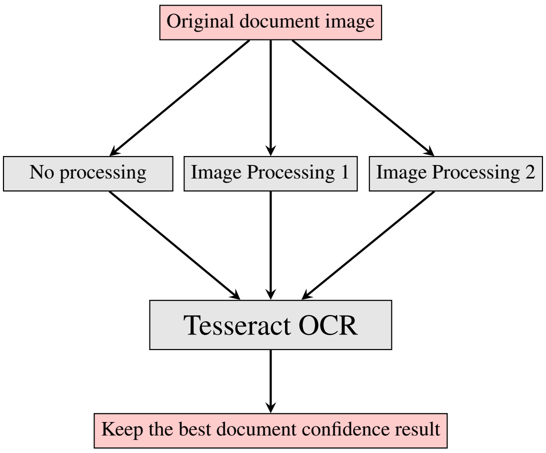

Figure 5: Proposed OCR method

4 Experiments

4.1 Data

For training the Finnish Fraktur model we used 72 images. The images contained 34 newspaper page images from our newspaper collection, and 38 synthetically created images with one column texts with various Finnish Fraktur fonts. The documents were manually boxed by using a box editor[10]. Training data contained 184,852 characters and two character combinations. Testing data was a proofread set of 462 scanned newspaper pages. It contained 530,579 word tokens, 4,175,710 characters. 434 pages were Fraktur, and the rest partly Fraktur and/or Antiqua pages.

4.2 Accuracy measure

As accuracy measures we use word error rate (WER) and character accuracy rate (CER)[11]. They are defined below in Equations 5 and 6.

$$CER = \frac{i + s + d}{n} \tag{5}$$

[8] https://kba.github.io/hocr-spec/1.2/

[9] https://github.com/tesseract-ocr/langdata/tree/master/fin

[10] https://github.com/zdenop/qt-box-editor

[11] http://www.digitisation.eu/training/succeed-training-materials/ocr-evaluation/ocrevaluation/measuring-ocr-quality/computing-error-rates/

where n is the total number of characters in reference text, i is the minimal number of insertions, s is substitutions and d is deletions on character level to obtain the reference text.

$$WER = \frac{i_w + s_w + d_w}{n_w} \tag{6}$$

where n_w is the total number of words in reference text, i_w is the minimal number of insertions, s_w is substitutions and d_w is deletions on word level to obtain the reference text. Smaller WER and CER means better quality for words and characters. For the WER results Impact centre's accuracy evaluation tool developed by Carrasco (2014) was utilized in this research. The Fraktur model CER results were compared by utilizing the Information Science Research Institute's OCR accuracy tool developed by Rice and Nartker (1996).

4.3 Results

WER results for the different runs of our method can be seen below in Table 1. Result of ABBYY FineReader 11 and 7 or 8 are shown as a baseline, our method with different image processing methods and an Oracle are given as comparison. In our method, Tesseract was run with two models (Fraktur and Antiqua), where Tesseract selects the better one as a result. Oracle is the best result of the combined 2-4 images based on the resulting WER.

Method	Result	Oracle
ABBYY FineReader 11	20.94	N/A
ABBYY FineReader 7 or 8	26.23	N/A
Tesseract (fi_frak_mk41+fin)		
original (otsu)	23.32	N/A
gaussian blur	24.24	N/A
sauvola	39.49	N/A
wolf	**22.67**	N/A
clahe	29.19	N/A
l.norm	23.36	N/A
clahe+wolf	32.67	N/A
l.norm+wolf	22.76	N/A
Combined 2		
l.norm+wolf,clahe+wolf	20.30	19.42
clahe+wolf,orig	20.40	19.30
clahe+wolf,blur	20.58	19.44
clahe+wolf,wolf	**19.98**	**19.19**
Combined 3		
clahe+wolf,wolf,blur	19.58	**17.53**
l.norm+wolf, clahe+wolf,blur	19.68	17.57
l.norm+wolf, clahe,orig	19.69	17.99
l.norm, clahe+wolf,wolf	**19.14**	17.69
Combined 4		
l.norm, clahe+wolf,orig,blur	19.11	17.61
l.norm, clahe+wolf,orig,wolf	19.32	**16.94**
l.norm+wolf, clahe+wolf,orig,blur	19.41	**16.94**
l.norm+wolf, clahe+wolf,orig,wolf	**19.02**	17.50

Table 1: WER results, the best of each 1-4 runs on bold (smaller is better)

CER results for the Fraktur model for original images can be seen below in Figure 3 and 4. The figures present the most frequent character errors, and their correctness percentage for the Fraktur model, respectively.

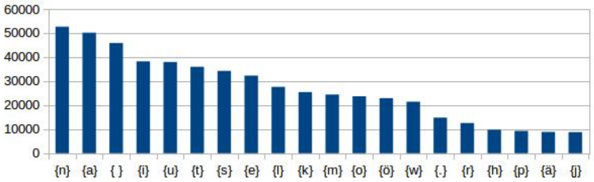

Figure 6: Most frequent character errors

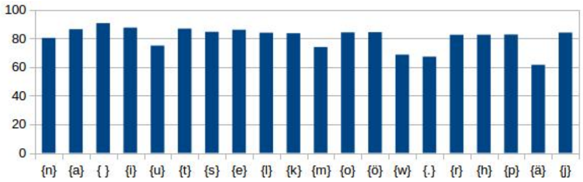

Figure 7: Correctness percentage

4.4 Execution times

Tesseract was run on IT Centre for Science (CSC)[12] machines on 8 core Intel 2.6GHz CPU, 2GB RAM, in Red Hat Enterprise Linux 7 (64-bit) virtual environment, in 8 document batches by GNU parallel, developed by Tange (2011). ABBYY FineReader 11 was run on 1 core Intel 2.6GHz CPU, 4GB RAM, in Windows Server (32-bit) virtual environment. Decoding speeds for the Tesseract and ABBYY FineReader were 722,323 tokens/hour and 30,626 tokens/hour, respectively. Re-OCRing millions of pages is feasible using multiple CSC machines in parallel.

[12] https://www.csc.fi

5 Conclusions

We have described a method for improving the Tesseract OCR toolkit for Finnish historical documents. In the tests Tesseract was run in dual model mode using created Finnish Fraktur model and Finnish Antiqua model, and selecting the best document by its confidence value with different 2-4 image combinations. This method has clearly outperformed the ABBYY FineReader results.

The best method was achieved by combining four methods (Linear Normalization + WolfJolion, Contrast Limited Adaptive Histogram Equalization+WolfJolion, original image and WolfJolion), which improves the word level quality of OCR by 1.91 percentage points (which is 9.16%) against the best result on FineReader 11 and by 7.21 percentage points (which is 27.48%) against the FineReader 7 and 8. It is great that FineReader 11 results have clearly improved from an earlier FineReader results. However, our current whole collection have been run only on FineReader 7 and 8, and the FineReader 11 is not feasible to be run for the whole collection due to the current licencing policy. Therefore, our method would correct about 84.6 million words (27.48%) more in the current 1.06 million Finnish newpaper page collection (containing Finnish language). However, it can still be improved. The method is 2.08 percentage points from the optimal Oracle result (16.94).

The character accuracy results for Fraktur model show that characters u, m and w have under 80 percent correctness. These letters are overlapping with letters such as n and i. It seems, however, that if accuracy for one of them is increased accuracy of others will decrease. Possibly also letter ä could be improved, though is has similar overlapping with letters a and å. From 20 most frequent errors only five are under 80% correct. Still, the Fraktur model could be developed more, possibly to recognize also bold letters.

Tesseract's document confidence value can be used to find the weak quality documents for further processing. However, it is not a perfect measure when comparing and/or combining other model results together. The confidence measure could possibly be improved by integrating it with a morphological tool, that checks words after OCR, and then weights the confidence measure for each word. The image quality is one of the most important factors in the recognition accuracy, so further research with image processing algorithms should continue. In addition to utilizing the confidence measure value, methods to determine noise level in the image could possibly be utilized to choose only bad quality images for further preprocessing.

Acknowledgments

This research was supported by European Regional Development Fund (ERDF), University of Helsinki, Mikkeli University Consortium and the City of Mikkeli. (Project Digitalia)

References

R. C. Carrasco. 2014. An open-source OCR evaluation tool. In *DATeCH '14 Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, pages 179–184.

M. Droettboom. 2003. Correcting broken characters in the recognition of historical documents. In *JCDL 03 Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, pages 364–366.

A. El Harraj and N. Raissouni. 2015. Ocr accuracy improvement on document images through a novel preprocessing approach. In *Signal & Image Processing : An International Journal (SIPIJ)*, volume 6, pages 114–133.

J. Evershed and K. Fitch. 2014. Correcting Noisy OCR: Context beats Confusion (2014). In *DATeCH '14 Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, pages 45–51.

G. Ganchimeg. 2015. History document image background noise and removal methods. In *International Journal of Knowledge Content Development & Technology*, volume 5, pages 11–24.

R. C. Gonzales and R. E. Woods. 2002. *Digital Image Processing*. Prentice-Hall.

M. Heliński, M. Kmieciak, and T. Parkoła. 2012. Report on the comparison of Tesseract and ABBYY FineReader OCR engines. Technical report, Poznań Supercomputing and networking center, Poland.

N. Howe. 2013. Document Binarization with Automatic Parameter Tuning. In *Journal International Journal on Document Analysis and Recognition*, volume 16, pages 247–258.

A. Järvelin, H. Keskustalo, E. Sormunen, M. Saastamoinen, and K. Kettunen. 2015. Information retrieval from historical newspaper collections in highly inflectional languages: A query expansion approach. In *Journal of the Association for Information Science and Technology*, volume 67.

K. Kettunen, T. Pääkkönen, and M. Koistinen. 2016. Between diachrony and synchrony: evaluation of lexical quality of a digitized historical Finnish newspaper collection with morphological analyzers. In *Baltic HLT 2016*, volume 289, pages 122–129.

R. Krutsch and D. Tenorio. 2011. Histogram Equalization, Application Note. Technical report.

D. Lopresti. 2009. Optical character recognition errors and their effects on natural language processing. In *International Journal on Document Analysis and Recognition*, volume 12, pages 141–151.

N. Makkar and S Singh. 2012. A Brief tour to various Skew Detection and Correction Techniques. In *International Journal for Science and Emerging Technologies with Latest Trend*, volume 4, pages 54–58.

W. Niblack. 1986. *An Introduction to Image Processing*, volume SMC-9. Prentice-Hall, Eaglewood Cliffs, NJ.

K. Ntirogiannis, B. Gatos, and I. Pratikakis. 2014. ICFHR2014 Competition on Handwritten Document Image Binarization (H-DIBCO 2014). In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 809–813.

N. Otsu. 1979. A Threshold Selection Method from Gray-Level Histograms. In *IEEE Transactions on Systems, Man and Cybernetics*, volume SMC-9, pages 62–66.

S. Parashar and S. Sogi. 2012. Finding skewness and deskewing scanned document. 3(4):1619–1924.

S. M. Pizer, R. E. Johnston, J. P. Ericksen, B. C. Yankaskas, and K. E. Muller. 1990. Contrast Limited Histogram Equalization Speed and Effectiveness.

I. Pratikakis, B. Gatos, and K. Ntirogiannis. 2013. ICDAR 2013 Document Image Binarization Contest (DIBCO 2013). In *2013 12th International Conference on Document Analysis and Recognition*, pages 1471–1476.

S. V. Rice and T. A. Nartker. 1996. The ISRI Analytic Tools for OCR Evaluation Version 5.1. Technical report, Information Science Research Institute (ISRI).

J. Sauvola and M. Pietikäinen. 1999. Adaptive Document Image Binarization. In *The Journal of the Pattern recognition society*, volume 33, pages 225–236.

M. Segzin and B. Sankur. 2004. Survey over image thresholding techniques and quantitative performance evaluation.

R. Smith. 1995. A Simple and Efficient Skew Detection Algorithm via Text Row Algorithm. In *Proceedings 3rd ICDAR'95, IEEE (1995)*, pages 1145–1148.

R. Smith. 2007. An Overview of the Tesseract OCR Engine. In *Proc. Ninth Int. Conference on Document Analysis and Recognition (ICDAR), IEEE (1995)*, pages 629–633.

M. L. Smitha, P. J. Antony, and D. N. Sachin. 2016. Document Image Analysis Using Imagemagick and Tesseract-ocr. In *International Advanced Research Journal in Science, Engineering and Technology (IARJSET)*, volume 3, pages 108–112.

T. Stanhope. 2016. Applications of Low-Cost Computer Vision for Agricultural Implement Feedback and Control.

O. Tange. 2011. GNU Parallel - The Command-Line Power Tool. In *The USENIX Magazine*, pages 42–47.

S. Tanner, T. Muñoz, and P. Hemy Ros. 2009. Measuring Mass Text Digitization Quality and Usefulness. Lessons Learned from Assessing the OCR Accuracy of the British Library's 19th Century Online Newspaper Archive. 15(7/8).

C. Wolf, J. Jolion, and F. Chassaing. 2002. Text Localization, Enhancement and Binarization in Multimedia Documents. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 4, pages 1037–1040. Quebec City, Canada.