

Image Text Enhancer

Algorithm Engineering 2026 Project Paper

Daniel Motz
Friedrich Schiller University Jena
Jena, Germany
daniel.motz@uni-jena.de

Leonard Teschner
Friedrich Schiller University Jena
Jena, Germany
leonard.teschner@uni-jena.de

Mher Mnatsakanyan
Friedrich Schiller University Jena
Jena, Germany
mher.mnatsakanyan@uni-jena.de

Abstract

The five-finger pattern:

- (1) **Topic and background:** What topic does the paper deal with? What is the point of departure for your research? Why are you studying this now?
- (2) **Focus:** What is your research question? What are you studying precisely?
- (3) **Method:** What did you do?
- (4) **Key findings:** What did you discover?
- (5) **Conclusions or implications:** What do these findings mean? What broader issues do they speak to?

Keywords

noise reduction, background removal, image filter, binarization

ACM Reference Format:

Daniel Motz, Leonard Teschner, and Mher Mnatsakanyan. 2026. Image Text Enhancer: Algorithm Engineering 2026 Project Paper. In *Proceedings of AEPRO 2026: Algorithm Engineering Projects (AEPRO 2026)*. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 Introduction

1.1 Background

In Zeiten der Digitalisierung werden viele gedruckte, handschriftliche und historische Dokumente mit einem Scanner oder der Kamera eines Smartphones digitalisiert. Dabei entstehen häufig Bilder in schlechter Qualität, die sich nicht für digitale Bildverarbeitungsverfahren wie die Texterkennung eignen [2, 20, 23]. Probleme bei der Digitalisierung können durch die Aufnahmeart sowie durch den Dokumententyp unterschiedliche Herausforderungen entstehen: Dokumente, die mit einer Smartphone-Kamera gescannt wurden, können nicht alle Details wie mit einem dedizierten Dokumentenscanner erfassen. Dadurch entstehen verschwommene und verzerrte Bilder. Auch der Winkel und der Abstand der Kamera zum Dokument beeinflussen die Qualität. So können verzerrte Perspektiven und Probleme beim Text-Alignment verursacht werden. Die Lichteigenschaften können Schatten werfen sowie Glare und Reflexionen hervorrufen. Zusätzlich können Smartphone-Kameras Text erfassen, der nicht zum gescannten Dokument selbst gehört [2, 5]. Gerade bei historischen Dokumenten besteht ein großes Interesse daran, sie zu digitalisieren. So bleiben sie erhalten und können mit Texterkennung analysiert werden. Aufgrund ihres hohen Alters, der Lagerung und der Art und Weise ihrer Erstellung leiden sie jedoch häufig unter Problemen. So können sie Fading und Noise aufweisen, wodurch der Text unlesbar wird. Die Digitalisierung wird durch handschriftliche Notizen, überlappende Texte,

stilistische Variationen sowie durch beschädigte Seiten, Risse und Schimmelstellen komplexer [5, 14].

1.2 Related Work

Bei der Bildverbesserung sowie bei der Analyse digitalisierter Dokumente mithilfe von OCR-Texterkennung und Recognition-Systemen ist die Segmentierung von Hintergrund und Vordergrund ein wichtiger Schritt [20]. Die Segmentierung wird mit der Binarisierung umgesetzt. Bei der Binarisierung gibt es traditionelle Methoden, welche einen globalen [12], lokalen [4, 6, 13] oder auch gemixten Threshold berechnen [19, 20] und pixelweise mit ihm die Pixel als Vorder- und Hintergrund klassifizieren. Mit Image Feature Methoden wurden Edge Detection [10, 20] und Fuzzy Logics [16, 20] für die Binarisierung eingesetzt. In den letzten Jahren sind zu den traditionellen Methoden zusätzlich Deep Learning Binarisierungsmethoden hinzugekommen. Diese beruhen auf Convolutional Neural Networks, Generative Adversarial Networks oder Attention Mechanisms. [20].

Neben den reinen Binarisierungsmethoden gibt es Ansätze, die mehrere Bildverarbeitungsmethoden kombinieren, um die Qualität von Textbildern zu verbessern. So stellt Alqudah et al. [2] eine Pipeline vor, die Entropiefilter und morphologische Operationen mit Binarisierung kombiniert. In [11] wird Niblack's Binarisierung mit einem Laplace Kantenfilter und globaler Optimierung kombiniert. Ein weiterer Ansatz wird von Vlasceanu et al. [17] beschrieben. Dabei werden verschiedene Binarisierungsmethoden kombiniert und mittels eines Voting-Mechanismus entschieden, welcher Pixel dem Vorder- bzw. Hintergrund zuzuordnen ist.

1.3 Our Contribution

In diesem Paper stellen wir eine Pipeline vor, die mehrere Bildverarbeitungsmethoden kombiniert, um die Qualität von gescannten oder fotografierten Textbildern zu verbessern. Sie umfasst Schritte wie Deskewing, Kontrastverbesserung, Rauschunterdrückung, Binarisierung, Despeckle, und morphologische Operationen. Das binarisierte Ergebnis der Pipeline kann außerdem zurück in ein Farbbild konvertiert werden.

Die Pipeline ist modular und benutzerfreundlich aufgebaut. Nutzer können auswählen, welche Schritte sie anwenden möchten, um die Pipeline an ihre spezifischen Anwendungsfälle anzupassen. Die Methodenparameter können einzeln angepasst werden, sind aber standardmäßig mit den State-of-the-Art-Werten konfiguriert, um die Benutzerfreundlichkeit zu gewährleisten.

Uns ist die effiziente Verarbeitung in der Pipeline wichtig. Daher vergleichen wir uns mit traditionellen open-source Binarisierungsmethoden und Bildverbesserungsverfahren und lassen bewusst Deep Learning Ansätze, sowie zahlungspflichtige Software außen vor.

Wir stellen die Pipeline als C++ Bibliothek, zusätzlich mit den einzelnen Methoden zur Verfügung. Die Implementierung nutzt Parallelisierung durch OpenMP, sowie weitere Optimierungsmethoden wie Loop-blocking um eine effiziente Verarbeitung großer Bilddatenmengen zu gewährleisten.

1.4 Outline

This paper is structured as follows: Section 2 provides a detailed description of the developed pipeline and its methods. Section 3 demonstrates the performance of our pipeline using experiments. Finally, Section 4 summarises our results and provides an outlook on possible future work.

2 The Pipeline

Many approaches and best practices already exist for improving the quality of scanned images as seen in section 1.2. We have developed a pipeline that combines several of these methods in order to achieve potentially good results. Users can choose which steps to apply from the pipeline. The individual methods of the pipeline are shown in algorithm 1.

Algorithm 1 Image Text Enhance Pipeline

- (1) convert image to grayscale
 - (2) Deskew (if requested)
 - (3) Contrast enhancement
 - (4) Denoising
 - (5) Binarization
 - (6) Despeckle (if requested)
 - (7) Morphological operations (if requested)
 - (8) Color passthrough
-

The individual methods of the pipeline are explained in more detail below.

2.1 Convert image to grayscale

All pipeline methods work on grayscale images. Therefore, the first step is to convert the input image into a grayscale image. This is achieved by applying the weighted sum $Y = 0.299R + 0.587G + 0.114B$, as defined by the International Telecommunication Union [9], to each pixel. The result is a grayscale image in which the brightness value of each pixel, represented by Y , corresponds to that of the original RGB-pixel. All steps in the pipeline are performed on the converted grayscale image in-place after the conversion.

2.2 Deskew

Damit der Text im Bild horizontal ausgerichtet ist, wird ein Deskew Schritt auf den Wunsch des Nutzers hin durchgeführt. Besonders Text-Analyse-Methoden wie OCR profitieren von horizontal ausgerichteten Texten [3]. Unser Deskew Algorithmus nutzt die Projection Profile Methode, die ähnlich zu [1] ist. Zuerst wird das Bild in Graustufen konvertiert und mittels Sauvola's Methode binarisiert¹. Im zweiten Schritt wird der Winkel gesucht, der die horizontale Projektions Varianz maximiert. Abschließend wird das Bild rotiert, um den Skew zu korrigieren.

¹siehe Abschnitt 2.5 oder [13]

Die Vorteile dieser Methode sind, dass sie Polarity-safe ist², eine coarse-to-fine Winkelsuche für Effizienz nutzt und Neumann Randbedingungen verwendet, um schwarze Ecken zu vermeiden.

2.3 Contrast enhancement

Um die unterschiedlichen Lichtverhältnisse und daraus entstandenen Kontrastprobleme bei der Digitalisierung von Dokumenten zu adressieren, wird ein Kontrastverbesserungsschritt durchgeführt. Dies hilft der Binarisierung im weiteren Verlauf [2]. Dabei wird ein robuster linearer Kontrast Stretching Algorithmus angewendet. Dabei werden die unteren 1% und oberen 1% der Intensitäten abgeschnitten, um Ausreißer zu ignorieren. Der verbleibende Bereich wird dann auf den vollen Bereich von 0 bis 255 gestreckt.

2.4 Denoising

Digitale Bilder sind durch ihre Aufnahmeart, Kompression oder durch ihren Transmissionchannel mit Rauschen versehen, bei dem Bildinformationen verloren gehen. Durch das Vorhandensein von Rauschen sind unter anderem Bildanalyse-Schritte in ihrer Effektivität eingeschränkt [7]. Daher wird in der Pipeline ein Denoising Schritt durchgeführt, der das Rauschen reduziert. Es stehen verschiedene einfache und adaptive Filtermethoden zur Verfügung, die der Nutzer auswählen kann. Es werden zwei einfache Filtermethoden angeboten, die von CImg bereitgestellt werden: Ein Gaussian Filter mit Neumann Randbedingungen und ein nicht-linearer Median Filter [15]. Der Gauß Filter ist Teil der Low Pass-Filter und zeichnet damit das Bild weich. Der Median Filter ist ein nicht-linearer Filter und nimmt den Pixelwert an, der den Median der Nachbarschaftspixel im Fenster darstellt [22]. Zusätzlich zu den beiden einfachen Filtern werden zwei adaptive Filtermethoden angeboten: Ein adaptiver Gaussian Blur Filter, der eine variable Standardabweichung nutzt, um an Kanten weniger zu verwischen und in flachen Regionen, mit hoher Varianz, stärker zu verwischen. Zusätzlich zum einfachen Median Filter wird ein adaptiver Median Filter bereitgestellt, der besonders gut für die Entfernung von Impulsrauschen (salt-and-pepper) geeignet ist, während Kanten und feine Details erhalten bleiben. Dabei wird mit einem 33 Fenster begonnen, das bei der Erkennung von Impulsrauschen bis zu einer definierten maximalen Fenstergröße erweitert wird, während nicht-Impuls-Pixel unverändert bleiben. Dies macht ihn ideal für die Entfernung von Scan-Speckle in Textbildern.

2.5 Binarization

Die Segmentierung von Vorder- und Hintergrund ist ein guter Ansatz, um die Bildqualität zu verbessern und stellt zudem den ersten Schritt bei Recognition Systemen wie OCR [2, 20]. Da die Binarisierungsmethoden in der Literatur gut untersucht sind, bieten wir mehrere Binarisierungsmethoden an, die der Nutzer auswählen kann. Alle Methoden sind Thresholding Methoden, die für jeden Pixel einen Threshold T berechnen (T_g für globale und T_w für lokale Thresholds) und den Pixelwert $i(x, y)$ mit dem Threshold vergleichen. Diese Art der Binarisierung ist simpel und effizient [5].

Die einfachste Methode ist die Otsu Methode [12], welche einen globalen Threshold berechnet. Dabei wird die intra-class Varianz minimiert. Dies ist in der Gleichung (1) und (2) dargestellt, wobei

²erkennt hellen vs. dunklen Hintergrund

$w1(t)$ und $w2(t)$ die Wahrscheinlichkeiten der zwei Klassen (Vorder- und Hintergrund) sind und $\sigma^2_1(t)$ und $\sigma^2_2(t)$ die Varianzen der zwei Klassen sind. Dies macht die Methode effizient, aber anfällig für Überlappungen und schlechte Intensitätsverteilungen [5].

$$\sigma^2_w(t) = w1(t) * \sigma^2_1(t) + w2(t) * \sigma^2_2(t) \quad (1)$$

$$T_g = \underset{t}{\operatorname{argmin}} \sigma^2_w(t) \quad (2)$$

Ein lokaler Treshold ansatz, der adaptiv ist, ist die Sauvola Methode [13]. Dabei wird für jedes Pixel ein lokaler Threshold T_w in einem Fenster um das Pixel berechnet. Diese Methode ist resistent gegen ungleichmäßige Beleuchtung. Es wird der lokale Mittelwert m_w und die lokale Standardabweichung σ_w des Fensters, sowie ein Parameter R , welcher die dynamische Range der Standardabweichung darstellt, genutzt. Mit dem Parameter k kann die Sensitivität des Thresholds korrigiert werden. Der Threshold wird in Gleichung (3) dargestellt [13, 20]. Wir haben noch den optionalen Paramter δ hinzugefügt, um den Threshold weiter feinjustieren zu können.

$$T_w = m_w * \left(1 + k * \left(\frac{\sigma_w}{R} - 1\right)\right) - \delta \quad (3)$$

Eine weitere adaptive Methode, die adaptive die benötigte Fenstergröße anpasst, ist die Methode von Bataineh et al. [4]. Diese Methode berechnet zuerst einen globalen Threshold T_{con} (4), der den Pixelwert in Vordergrund (black), Hintergrund (weiß) und Confusion-Values (red) klassifiziert (5). Basierend auf dem Verhältnis von Vordergrund zu Confusion-Values und der globalen Standardabweichung wird eine primäre Fenstergröße PW_{size} ausgewählt (6). Sollte die Anzahl der Confusion-Values die der Vordergrundpixel in dem Fenster übersteigen, wird die halbe Fenstergröße SW_{size} verwendet.

$$T_{con} = m_g - \frac{m_g^2 * \sigma_g}{(m_g + \sigma_g) * (0.5max_{level} + \sigma_g)} \quad (4)$$

$$I = \begin{cases} \text{black,} & i(x, y) \leq T_{con} - \left(\frac{\sigma_g}{2}\right), \\ \text{red,} & T_{con} - \left(\frac{\sigma_g}{2}\right) < i(x, y) < T_{con} + \left(\frac{\sigma_g}{2}\right), \\ \text{white,} & i(x, y) \geq T_{con} + \left(\frac{\sigma_g}{2}\right), \end{cases} \quad (5)$$

$$PW_{size} = \begin{cases} \left(\frac{I_h}{4}, \frac{I_w}{6}\right), & \geq 2.5 \text{ or } (\sigma_g < 0.1 * max_{level}), \\ \left(\frac{I_h}{30}, \frac{I_w}{20}\right), & 1 < p < 2 - 5 \text{ or } (I_h + I_w < 400), \\ \left(\frac{I_h}{40}, \frac{I_w}{30}\right), & p \leq 1, \end{cases} \quad (6)$$

Anschließend wird für jedes Fenster der lokale Threshold T_w berechnet (7). Der Threshold nutzt einen adaptiven Standardabweichungswert $\sigma_{adaptive}$, der auf den maximalen und minimalen Wert der Standardabweichung aller Fenster basiert (8) [4]. Durch die adaptive Fenstergröße und den adaptiven Thresholdwert basierend auf den Bildmerkmalen, ist diese Methode robust gegenüber unterschiedlichen Herausforderungen wie dünnen Stiftstrichen und kontrastarmen Bildern, es bleibt unvermeidlich ein übermäßiger Hintergrund erhalten [20].

$$T_w = m_w - \frac{m_w^2 - \sigma_w}{(m_g + \sigma_w) * (\sigma_{adaptive} + \sigma_w)} \quad (7)$$

$$\sigma_{adaptive} = \frac{\sigma_w - \sigma_{min}}{\sigma_{max} - \sigma_{min}} \quad (8)$$

2.6 Despeckle

Um kleine Flecken, die bei der Binarisierung entstehen bzw. übrigbleiben zu entfernen, wird ein Despeckle Schritt angeboten. Dabei werden kleine zusammenhängende Komponenten (speckles) aus dem binarisierten Bild in-place entfernt. Komponenten, die kleiner als ein definierter Schwellenwert sind, werden entfernt. Für die detektion der zusammenhängenden Komponenten wird die `get_label()` Methode aus der CImg Bibliothek verwendet[15]. Diese berechnet die zusammenhängenden Komponenten mittels Hesselinks et al. Algorithmus [8].

2.7 Morphological operations

Nach der Segmentierung von Vorder- und Hintergrund können kleine Löcher oder Inseln entstehen, welche durch Opening und Closing Operationen entfernt werden können [21]. Wir bieten in unserer Pipeline die Möglichkeit, die morphologischen Operationen Dilation und Erosion anzuwenden. Dilation erweitert helle (weiße) Bereiche. Bei binären Bildern kann dies gebrochene Zeichen verbinden oder Striche verdicken. Erosion verkleinert helle Bereiche (erweitert dunkle Bereiche). Bei binären Bildern kann dies kleine Rauschpunkte entfernen oder Striche dünner machen [18, 21].

2.8 Color passthrough

Im letzten Schritt der Pipeline kann das binarisierte Bild genutzt werden, um die Farbwerte des Originalbildes zu erhalten. Dazu wird das binarisierte und verbesserte Bild als Maske verwendet. Alle Pixel, die im binarisierten Bild als Vordergrund (schwarz) klassifiziert wurden, werden durch die Farbe des darunterliegenden Pixels im Originalbild ersetzt. In Gleichung (9) beschreibt $I'(x, y, z)$ das Ergebnis, $I_{original}(x, y, z)$ das farbige Originalbild und $i(x, y)$ das binarisierte Bild.

$$I'(x, y, z) = \begin{cases} I_{original}(x, y, z), & \text{if } i(x, y) == \text{schwarz,} \\ \text{white,} & \text{else} \end{cases} \quad (9)$$

3 Experiments

4 Conclusions

References

- [1] Teruo Akiyama and Norihiro Hagita. 1990. Automated entry system for printed documents. *Pattern Recognition* 23, 11 (1990), 1141–1154. doi:10.1016/0031-3203(90)90112-X
- [2] Musab Kasim Alqudah, Mohammad F. Bin Nasrudin, Bilal Bataineh, Masha Alqudah, and Arwa Alkhatatneh. 2015. Investigation of binarization techniques for unevenly illuminated document images acquired via handheld cameras. In *2015 International Conference on Computer, Communications, and Control Technology (I4CT)*, 524–529. doi:10.1109/I4CT.2015.7219634
- [3] Wuzhida Bao, Cihui Yang, Shiping Wen, Mengjie Zeng, Jianyong Guo, Jingting Zhong, and Xingmiao Xu. 2022. A Novel Adaptive Deskewing Algorithm for Document Images. *Sensors* 22, 20 (2022). doi:10.3390/s22207944

- [4] Bilal Bataineh, Siti Abdullah, Khairuddin Omar, and Mohammad Faizul Nasrudin. 2011. Adaptive Thresholding Methods for Documents Image Binarization, Vol. 6718. 230–239. doi:10.1007/978-3-642-21587-2_25
- [5] Bilal Bataineh, Mohamed Tounsi, Nuha Zamzami, Jehan Janbi, Waleed Abdel Karim Abu-ain, Tarik AbuAin, and Shaima Elnazer. 2025. A Comprehensive Review on Document Image Binarization. *Journal of Imaging* 11, 5 (2025). doi:10.3390/jimaging11050133
- [6] Derek Bradley and Gerhard Roth. 2007. Adaptive Thresholding using the Integral Image. *J. Graphics Tools* 12 (01 2007), 13–21. doi:10.1080/2151237X.2007.10129236
- [7] Linwei Fan, Fan Zhang, Hui Fan, and Caiming Zhang. 2019. Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art* 2, 1 (2019), 7. doi:10.1186/s42492-019-0016-7
- [8] Wim H. Hesselink, Arnold Meijster, and Coenraad Bron. 2001. Concurrent determination of connected components. *Science of Computer Programming* 41, 2 (2001), 173–194. doi:10.1016/S0167-6423(01)00007-7
- [9] International Telecommunication Union. 2011. *Recommendation ITU-R BT.601-7: Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios*. Recommendation BT.601-7. ITU Radiocommunication Sector (ITU-R), Geneva, Switzerland. https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.601-7-201103-I!!PDF-E.pdf
- [10] Shijian Lu, Bolan Su, and Chew Lim Tan. 2010. Document image binarization using background estimation and stroke edges. *International Journal on Document Analysis and Recognition (IJ DAR)* 13, 4 (2010), 303–314. doi:10.1007/s10032-010-0130-8
- [11] Sergey Milyaev, Olga Barinova, Tatiana Novikova, Pushmeet Kohli, and Victor Lempitsky. 2013. Image Binarization for End-to-End Text Understanding in Natural Images. In *2013 12th International Conference on Document Analysis and Recognition*. 128–132. doi:10.1109/ICDAR.2013.33
- [12] Nobuyuki Otsu. 1979. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 9, 1 (1979), 62–66. doi:10.1109/TSMC.1979.4310076
- [13] J. Sauvola and M. Pietikäinen. 2000. Adaptive document image binarization. *Pattern Recognition* 33, 2 (2000), 225–236. doi:10.1016/S0031-3203(99)00055-2
- [14] Alaa Sulaiman, Khairuddin Omar, and Mohammad F. Nasrudin. 2019. Degraded Historical Document Binarization: A Review on Issues, Challenges, Techniques, and Future Directions. *Journal of Imaging* 5, 4 (2019). doi:10.3390/jimaging5040048
- [15] Jean-Philippe Tarel. 2024. CImg: A Simple C++ Toolkit for Image Processing. <https://cimg.eu>. Version 3.x.
- [16] Li-Jing Tong, Kan Chen, Yan Zhang, Xiao-Ling Fu, and Jian-Yong Duan. 2009. Document Image Binarization Based on NFCM. In *2009 2nd International Congress on Image and Signal Processing*. 1–5. doi:10.1109/CISP.2009.5305330
- [17] Giorgiana Violeta Vlăsceanu, Caraman Ghenadie, Răzvan Nițu, and Costin-Anton Boiangiu. 2022. A voting method for image binarization of text-based documents. In *2022 21st RoEduNet Conference: Networking in Education and Research (RoEduNet)*. 1–5. doi:10.1109/RoEduNet57163.2022.9921086
- [18] Martin Werner. 2020. *Digitale Bildverarbeitung*. Springer Vieweg Wiesbaden. doi:10.1007/978-3-658-22185-0
- [19] You Yang. 2008. OCR Oriented Binarization Method of Document Image. In *2008 Congress on Image and Signal Processing*, Vol. 4. 622–625. doi:10.1109/CISP.2008.262
- [20] Zhengxian Yang, Shikai Zuo, Yanxi Zhou, and Jianwen He, Jinlong and Shi. 2024. A Review of Document Binarization: Main Techniques, New Challenges, and Trends. *Electronics* 13, 7 (2024). doi:10.3390/electronics13071394
- [21] Yujin Zhang and Tsinghua University Press. 2017. *Image Engineering. Vol 2, Image Analysis* (1st ed. ed.). De Gruyter, Berlin ;.
- [22] Yu-Jin Zhang. 2017. *Image engineering*. De Gruyter, Berlin.
- [23] Yanxi Zhou, Shikai Zuo, Zhengxian Yang, Jinlong He, Jianwen Shi, and Rui Zhang. 2023. A Review of Document Image Enhancement Based on Document Degradation Problem. *Applied Sciences* 13, 13 (2023). doi:10.3390/app13137855