

Image Enhancement: A Lightweight, Easy-to-Use, and Efficient C++ Library

Algorithm Engineering 2026 Project

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

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

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

Abstract

TBC

Keywords

task parallelism, WebAssembly, OpenMP, image processing, C++

ACM Reference Format:

Daniel Motz, Leonard Teschner, and Mher Mnatsakanyan. 2026. Image Enhancement: A Lightweight, Easy-to-Use, and Efficient C++ Library: Algorithm Engineering 2026 Project. In *Proceedings of APRO 2026: Algorithm Engineering Projects (APRO 2026)*. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/nnnnnnnnnnnnnn>

1 Introduction

Possible Research Questions

- (1) Easily extensible Open Source Pipeline, ready to use out of the box with default options giving a perfectly good image, completely written in C++ and licensed under MIT-0 and CeCILL-C, which are both commercially available (compared to e.g. unpaper) and has no big dependencies (unlike unpaper, which uses ffmpeg) and only uses Clmg, which is portable (uses OpenMP), simple, self-contained, and lightweight.

1.1 Background

The execution of performance-critical applications in web browsers has changed with the introduction of WebAssembly (WASM). Although WASM offers near-native execution performance, utilising multi-core architectures with parallel programming is still a relatively new field. While native C++ programs use OpenMP to easily implement parallelism, support for OpenMP in WASM remains uncertain.

Was ist WASM?

Was machen wir mit unserer Pipeline?

- (1) easy to use
- (2) out of the box good results
- (3) low dependencies (only Clmg (doesnt have other dependencies) and OpenMP)
- (4) only image enhancement, no OCR
- (5) MIT-0 or CeCILL-C license (commercially usable)
- (6) modular (users can choose which steps to apply)

In this paper, we examine whether and how well OpenMP-based parallelism is compatible with WebAssembly by using a lightweight image enhancement pipeline implemented in C++ with OpenMP. First, we demonstrate that the pipeline can be compiled for WASM,

then we evaluate its performance in native execution and in WebAssembly. We then compare the results.

1.2 Related Work

Image Enhancement und Dokumenten Preprocessing wurden von einer weitreichenden Anzahl an open-source Tools und kommerzieller Software abgedeckt. Sie reichen von umfassenden Bildbearbeitungsbibliotheken, Object Character Recognition OCR Bibliotheken, bis hin zu spezialisierten Tools für die Nachbearbeitung von gescannten Dokumenten.

Eine der weitverbreitesten Bibliotheken für allgemeine Bildverarbeitung ist OpenCV. OpenCV wurde von Bradski entwickelt und bietet eine Vielzahl von Funktionen und Algorithmen für die Bildverarbeitung, beispielsweise Filter, Transformationen, Segmente runt und Feature-Erkennung, sowie Machine-und Deep-learning. OpenCV ist in C++ geschrieben und bietet eine modulare, Plattformübergreifende Architektur, die es einfach macht, es in verschiedenen Anwendungen zu integrieren. Allerdings ist OpenCV eine sehr umfassende Bibliothek, die viele Funktionen und Abhängigkeiten enthält, was zu einem Overhead führt, wenn nur ein Bruchteil der Funktionen genutzt wird. Zudem ist OpenCV unter der BSD-Lizenz lizenziert. Ähnlich zu OpenCV bietet ImageMagick eine umfassende Suite für die Bildbearbeitung, als Kommandozeilen-Tool und als Bibliothek. Es bietet Funktionen wie Formatkonvertierung, Größenänderung, Filterung und Farbkorrektur. Es ist jedoch nicht speziell auf die Verbesserung von gescannnten oder fotografierten Bildern ausgerichtet, sondern eher auf allgemeine Bildbearbeitung. Die Vielzahl an Funktionen und Möglichkeiten resultieren in einer Vielzahl an Abhängigkeiten, und einer großen Codebase. Es ist unter der Apache-Lizenz lizenziert.

Bei der Dokumenten-Analyse wird häufig die open-source OCR engine Tesseract verwendet. Tesseract wurde von Smith in 2007 beschrieben und hat sich in eine der führenden OCR-Engines entwickelt. Es bietet neben der Texterkennung auch einige Funktionen zur Bildverbesserung, dessen primärer Fokus jedoch in der OCR liegt. Die Bildverbesserungsfunktionen sind typischerweise in den OCR-Workflow integriert und stehen nicht als allgemeine Bildverbesserungsfunktionen zur Verfügung. Tesseract ist unter der Apache-Lizenz lizenziert. Eine ähnliche Bibliothek, die sich auf OCR spezialisiert hat, ist die C basierte Leptonica Bibliothek. Leptonica bietet Funktionen für die Bildverarbeitung, die häufig im Backend von OCR-Engines wie Tesseract verwendet werden. Es bietet Funktionen wie Binarisierung, Rauschunterdrückung und Morphologische Operationen, die für die Vorbereitung von Bildern für die OCR-Erkennung wichtig sind. Obwohl Leptonica relativ lightweight im

vergleich zu den anderen Bibliotheken ist, dient sie hauptsächlich als low-level Bibliothek und bietet beispielsweise keine ready-to-use konfigurierbare verarbeitungs Pipelines an. Außerdem ist sie in C geschrieben, was die Integration in C++ Anwendungen erschwert. Leptonica ist unter der BSD-Lizenz lizenziert.

Eine mehr spezialisiertes Tool ist unpaper, das sich auf die Nachbearbeitung von gescannten Dokumenten spezialisiert hat. Es bietet Funktionen wie Deskewing, Binarisierung und Rauschunterdrückung, die speziell für die Verbesserung von gescannten Dokumenten entwickelt wurden. Unpaper ist jedoch stark spezialisiert und bietet nicht die Flexibilität, um es in custom C++ Anwendungen zu integrieren. Basierend auf der Konfiguration benötigt unpaper viele Abhängigkeiten, was die Nutzung erschwert. Unpaper ist unter der GPL-Lizenz lizenziert, was die kommerzielle Nutzung einschränkt.

1.3 Our Contribution

Im Kontrast zu den beschriebenen, existierenden Tools und Bibliotheken, ist unser Ansatz darauf ausgelegt, minimalistisch, erweiterbar und einfach zu nutzen zu sein. Anstatt eine umfassende Bildverarbeitungsbibliothek oder eine OCR-Engine zu sein, konzentrieren wir uns auf eine lightweight, modulare Image Enhancement Pipeline, welche ausschließlich in C++ geschrieben ist und nur eine Abhängigkeit (CImg) hat, die selbst leichtgewichtig und portable ist. Durch die Verwendung von multithreading mit OpenMP können wir die Pipeline effizient ausführen und so auch große Bilder schnell verarbeiten. Unsere Arbeit unterscheidet sich von OpenCV und ImageMagick, durch die bewusste Einschränkung auf die Bildverbesserung und durch die kleingeblatene Architektur. Im Gegensatz zu Tesseract und Leptonica, ist unsere Pipeline nicht in einen OCR-Workflow integriert, und ist nicht durch low-level Primitive Funktionen beschränkt, sondern bietet eine konfigurierbare, standalone, ready-to-use Pipeline. Unsere Pipeline ist nicht ausschließlich auf die Nachbearbeitung von gescannten Dokumenten spezialisiert, sondern ist als allgemeine Image Enhancement Pipeline konzipiert, die für eine Vielzahl von Anwendungen genutzt werden kann. Zusätzlich sichert unser Lizenzmodell (MIT-0 oder CeCILL-C) die Nutzung in der Forschung sowie in kommerziellen Anwendungen. Durch die kombinierung von minimalistischen Abhängigkeiten, einer modularen Architektur, portabilität durch C++ und OpenMP, sowie einer kommerziell nutzbaren Lizenz, füllt unsere Pipeline eine Lücke in zwischen large-scale general-purpose Bildverarbeitungsbibliotheken und hochspezialisierten Tools für die Dokumentenverbesserung.

The pipeline is provided both as a executable and as a C++ library containing the individual methods. To efficiently process large amounts of image data, the implementation uses parallelization via OpenMP and other optimization methods such as loop blocking. The CImg library [1] is used for the basic image processing methods.

1.4 Outline

This paper is structured as follows: Section 2 provides a detailed description of the developed pipeline and its methods. Section 4 demonstrates the performance of our pipeline using experiments. Finally, Section 5 summarises our results and provides an outlook on possible future work.

2 The Pipeline

3 Pipeline Optimizations

4 Experiments

5 Conclusions

References

- [1] Jean-Philippe Tarel. 2024. CImg: A Simple C++ Toolkit for Image Processing. <https://cimg.eu>. Version 3.x.