UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ACCELERATE DOCUMENT PROCESSING WITH ARTIFICIAL INTELLIGENCE OBJECT CHARACTER RECOGNITION SOFTWARE

Diploma Thesis

Konstantinos Vakalis

Supervisor: Manolis Vavalis

July 2025

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ACCELERATE DOCUMENT PROCESSING WITH ARTIFICIAL INTELLIGENCE OBJECT CHARACTER RECOGNITION SOFTWARE

Diploma Thesis

Konstantinos Vakalis

Supervisor: Manolis Vavalis

July 2025

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΠΙΤΑΧΥΝΣΗ ΕΠΕΞΕΡΓΑΣΙΑΣ ΕΓΓΡΑΦΩΝ ΜΕ ΛΟΓΙΣΜΙΚΟ ΑΝΑΓΝΩΡΙΣΗΣ ΧΑΡΑΚΤΗΡΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

Διπλωματική Εργασία

Κωνσταντίνος Βακάλης

Επιβλέπων: Μανόλης Βάβαλης

Ιούλιος 2025

Approved by the Examination Committee:

Supervisor **Manolis Vavalis**
Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Elias N. Houstis**
Emeritus Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **George Thanos**
Laboratory Teaching Staff, Department of Electrical and Computer Engineering, University of Thessaly

Acknowledgements

I would like to express my heartfelt gratitude to my supervisor, Manolis Vavalis, for his invaluable guidance, and unwavering support throughout this journey. His insights and encouragement have been instrumental in my success. I am also deeply thankful to my family for their constant love, patience, and encouragement, which have given me the strength to persevere. Your support means the world to me.

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

«Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism».

The declarant

Konstantinos Vakalis

Diploma Thesis

ACCELERATE DOCUMENT PROCESSING WITH ARTIFICIAL INTELLIGENCE OBJECT CHARACTER RECOGNITION SOFTWARE

Konstantinos Vakalis

Abstract

Digitizing complicated legal documents is very hard since they have a combination of printed and handwritten text, complicated layouts, and specialist language. Strict data privacy and confidentiality rules sometimes make it impossible to use cloud-based solutions. This diploma thesis tackles this issue by suggesting, building, and testing a new, lightweight, on-premise OCR pipeline that is made to digitize Greek legal documents with high fidelity.

The methodology is based on a modular, multi-stage structure that smartly organizes and improves the output of existing OCR engines that have been tested and found to operate. A semantic segmentation model, derived on an adapted DeepLabv3+ architecture, initially distinguishes printed text, handwritten comments, and signatures. Then, these segments are smartly sent to a new multi-engine framework called TriOCR, which controls outputs from the high-accuracy commercial engine ABBYY FineReader, a transformer-based model (TrOCR) for handwriting, and the open-source Tesseract engine. Sophisticated alignment algorithms and a proprietary confidence-scoring system are used to create a strong agreement. Finally, a smart post-processing layer uses a custom-built Greek legal lexicon to fix errors that are specific to a certain field and make the text far more accurate overall.

We tested a carefully chosen set of complicated Greek legal documents to determine a quantitative performance standard. This test showed a big difference in performance between the two solutions. ABBYY FineReader had a Word Error Rate (WER) of 1.39%, whereas Tesseract had a WER of 33.01%. The suggested Tri-OCR pipeline shows that it is possible to make a secure, accurate, and useful solution that works better than individual engines and fits within the privacy and resource limits of the legal profession by smartly combining specialized engines and using domain-specific post-processing.

Keywords:

Optical Character Recognition, Document Processing, Artificial Intelligence, Legal Tech-

nology, Semantic Segmentation, Multi-Engine OCR, On-Premise OCR, Greek Legal Documents.

Διπλωματική Εργασία

ΕΠΙΤΑΧΥΝΣΗ ΕΠΕΞΕΡΓΑΣΙΑΣ ΕΓΓΡΑΦΩΝ ΜΕ ΛΟΓΙΣΜΙΚΟ ΑΝΑΓΝΩΡΙΣΗΣ ΧΑΡΑΚΤΗΡΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

Κωνσταντίνος Βακάλης

Περίληψη

Η ψηφιοποίηση σύνθετων νομικών εγγράφων αποτελεί μια σημαντική πρόκληση, η οποία χαρακτηρίζεται από μικτό περιεχόμενο (έντυπο και χειρόγραφο κείμενο), περίπλοκες διατάξεις και εξειδικευμένη ορολογία, ενώ διέπεται από αυστηρές απαιτήσεις προστασίας δεδομένων και εμπιστευτικότητας που συχνά αποκλείουν τη χρήση λύσεων που βασίζονται στο cloud. Η παρούσα διπλωματική εργασία αντιμετωπίζει αυτή την πρόκληση προτείνοντας, υλοποιώντας και αξιολογώντας μια καινοτόμο, ελαφριά και τοπικής εκτέλεσης (on-premise) αλυσίδα επεξεργασίας OCR, σχεδιασμένη για την υψηλής πιστότητας ψηφιοποίηση ελληνικών νομικών εγγράφων.

Η μεθοδολογία επικεντρώνεται σε ένα αρθρωτό πλαίσιο πολλαπλών σταδίων που συντονίζει και βελτιώνει έξυπνα την έξοδο υφιστάμενων, δοκιμασμένων μηχανών OCR. Ένα μοντέλο σημασιολογικής τμηματοποίησης, βασισμένο σε μια τροποποιημένη αρχιτεκτονική DeepLabv3+, διαχωρίζει αρχικά το έντυπο κείμενο από τις χειρόγραφες σημειώσεις και τις υπογραφές. Τα τμήματα αυτά δρομολογούνται στη συνέχεια σε ένα καινοτόμο πλαίσιο πολλαπλών μηχανών, ονόματι TriOCR, το οποίο ενορχηστρώνει τις εξόδους από την υψηλής ακρίβειας εμπορική μηχανή ABBYY FineReader, ένα μοντέλο βασισμένο σε transformers (TrOCR) για το χειρόγραφο κείμενο, και τη μηχανή ανοιχτού κώδικα Tesseract. Μια στιβαρή συναίνεση επιτυγχάνεται μέσω προηγμένων αλγορίθμων στοίχισης και ενός προσαρμοσμένου μηχανισμού βαθμολόγησης εμπιστοσύνης. Τέλος, ένα εξελιγμένο επίπεδο μετεπεξεργασίας αξιοποιεί ένα ειδικά κατασκευασμένο ελληνικό νομικό λεξικό για τη διόρθωση σφαλμάτων που αφορούν την εξειδικευμένη ορολογία και τη βελτίωση της συνολικής ακρίβειας.

Η πειραματική αξιολόγηση πραγματοποιήθηκε σε ένα επιμελημένο σύνολο δεδομένων από σύνθετα ελληνικά νομικά έγγραφα, θέτοντας ένα ποσοτικό σημείο αναφοράς απόδοσης. Αυτή η συγκριτική αξιολόγηση ανέδειξε ένα σημαντικό χάσμα απόδοσης μεταξύ των υφιστάμενων λύσεων, με την ABBYY FineReader να επιτυγχάνει Ποσοστό Λάθους Λέξεων (WER)

1.39% έναντι 33.01% του Tesseract. Η προτεινόμενη αλυσίδα επεξεργασίας TriOCR αποδεικνύει ότι, μέσω της έξυπνης ενσωμάτωσης εξειδικευμένων μηχανών και της εφαρμογής μετεπεξεργασίας προσαρμοσμένης στον τομέα, είναι εφικτή η δημιουργία μιας ασφαλούς, ακριβούς και πρακτικής λύσης που υπερτερεί των μεμονωμένων συστημάτων και λειτουργεί αποτελεσματικά εντός των περιορισμών ιδιωτικότητας και πόρων του νομικού επαγγέλματος.

Λέξεις-κλειδιά:

Οπτική Αναγνώριση Χαρακτήρων, Επεξεργασία Εγγράφων, Τεχνητή Νοημοσύνη, Νομική Τεχνολογία, Σημασιολογική Τμηματοποίηση, OCR Πολλαπλών Μηχανών, Τοπική Εκτέλεση OCR, Ελληνικά Νομικά Έγγραφα.

Table of contents

List of figures

List of tables

Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
AWS	Amazon Web Services
BPF	Band Pass Filter
BPPTS	Balanced Prediction Priority Task Scheduling
bWER	Bag of Words Word Error Rate
CC	Connected Component
CER	Character Error Rate
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CRAFT	Character Region Awareness for Text detection
CRNN	Convolutional Recurrent Neural Network
CSV	Comma-Separated Values
CTC	Connectionist Temporal Classification
CTPN	Connectionist Text Proposal Network
DMS	Document Management System
DRL	Deep Reinforcement Learning
EAST	Efficient and Accurate Scene Text Detector
FLOPs	Floating-point Operations
FSP	Flexible Synchronous Parallel
GDPR	General Data Protection Regulation
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HIPAA	Health Insurance Portability and Accountability Act
IMR	Intelligent Machines Research Corporation

IRIS	Image Recognition Integrated Systems
JSON	JavaScript Object Notation
LAMBERT	Layout-Aware Language Modeling for Information Extraction
LIFE	Local Information Enhancer
LRU	Least Recently Used
LSTM	Long Short-Term Memory
MCMC	Multi-Constrained Model Compression
MSER	Maximally Stable Extremal Regions
NCS	Neural Compute Stick
non-IID	Non-identically distributed
NSFD	Normalized Spearman's Foot Rule Distance
OCR	Optical Character Recognition
OIWHO	Improved Wild Horse Optimization
PDF	Portable Document Format
PLM	Pre-trained Language Model
PSO	Particle Swarm Optimization
PSM	Page Segmentation Mode
PTA	Text-aware Patch Alignment
RAM	Random-Access Memory
RNN	Recurrent Neural Network
ROI	Region of Interest
RPN	Region Proposal Network
SAC	Soft Actor-Critic
SDK	Software Development Kit
SVM	Support Vector Machine
SWT	Stroke Width Transform
TFLite	TensorFlow Lite
TFS-ViT	Token-level Feature Stylization Vision Transformer
TIFF	Tagged Image File Format
TPU	Tensor Processing Unit
TrOCR	Transformer Optical Character Recognition
VGT	Vision Grid Transformer

ViT	Vision Transformer
VQA	Visual Question Answering
WER	Word Error Rate
wmAP	Weighted Mean Average Precision
XML	Extensible Markup Language
YOLO	You Only Look Once

Chapter 1

Introduction

In today's environment of digital transformation, organizations across industries are faced with a rapidly increasing number of documents for which efficient management, effective archiving, and the mining for insights are a pressing need. Optical Character Recognition (OCR) technology has become the anchor technology for this transition, providing the bridge between physical or image-based information and machine-understandable, actionable data. Fueled by the strength of artificial intelligence and deep learning, state-of-the-art OCR technology achieved remarkable effectiveness and frequently creates the impression of text recognition as a ”solved problem”. As a result, however, for the most part, such a perception holds true only for standardized, high-grade printed documents. The big gap still remaining between performance under ideal conditions and the challenges present under real-world and special application conditions encompasses the challenges for the processings of degraded or aged documents, the varying and intricate layouts with tables and mixed contents, and the subtle issue of machine reading text that was hand-written. Furthermore, when applied for high-stakes applications such as the legal arena, the accompanying technical challenges are fortified by stringent operating constraints, such as the timeliness for immediate proceeding and the absolute requirements for the secrecy and the security of the data, often refusing the option for cloud-based application. The thesis explores this problematic nexus, investigating the challenges that emerge when preparing and releasing advanced OCR technology on highly complex documents under conditions of strict confidentiality and resource constraints.

1.1 Subject of the thesis

The diploma thesis directly tackles the problem of high-fidelity digitalization of papers within the law sector, which is limited by a few technical constraints and strict laws on privacy. The overall task for the project is creating a mechanism for a local system to process quickly and accurately complex Greek law papers into machine-readable form using a minimum amount of resources. These papers often include a mixture of print and handwriting, sophisticated multi-column layout, tables, and technical jargon. It's the only manner it can be accomplished since the papers are confidential and laws like the GDPR and the notion of the attorney-client privilege inhibit the employment of cloud-based processor programs.

This thesis proposes the design, implementation, and assessment of a lightweight, AI-powered OCR pipeline to address these challenges. This work is built on a modular, multi-stage architecture that intelligently coordinates and improves the output of existing, well-tested OCR engines. It does not develop a single, huge deep learning model from scratch. The suggested solution is based on a few key parts:

- A powerful preprocessing module that solves problems that often happen in scanned legal files, like skew and noise, by normalizing and enhancing photos of damaged documents.
- A clever semantic segmentation system that employs a modified DeepLabv3+ architecture to tell the difference between printed text, handwritten notes, and signatures at a very fine level.
- TriOCR is a novel multi-engine framework that brings together and organizes outputs from three different OCR engines: ABBYY FineReader, a commercial engine with high accuracy; TrOCR, a transformer-based model for handwriting; and Tesseract, an open-source engine. This framework uses advanced alignment algorithms and a special confidence-scoring method to come to a solid agreement.
- A complicated layer of post-processing that uses a custom-built Greek legal lexicon and a voting system based on consensus to rectify mistakes that are unique to a single field and make the text far more accurate overall.

The objective of this thesis is to demonstrate that the astute amalgamation of current technologies into a novel, modular pipeline can yield a secure, accurate, and pragmatic solution for the

digitization of legal documents that transcends standalone engines and operates effectively within the privacy and resource constraints of the legal profession.

1.1.1 Contribution

The main points of this thesis are as follows:

1. **Set up a quantitative performance test:** We did a full benchmark of the best commercial (ABBYY FineReader, Adobe Acrobat, IRIS) and open-source (Tesseract) OCR systems on a carefully chosen set of Greek legal documents. This investigation offers the inaugural quantitative proof for this area, delineating a distinct performance hierarchy and exposing a substantial accuracy disparity, with ABBYY FineReader attaining a Word Error Rate (WER) of 1.39% in contrast to Tesseract's 33.01%.

2. **Created an Intelligent Segmentation Pipeline:** To solve the problem of documents with heterogeneous content, we used and changed a semantic segmentation model based on the DeepLabv3+ architecture. This system can tell the difference between printed text and handwritten text. This lets each part be sent to the best OCR engine for specialized processing.

3. **Created and put into use a new multi-engine OCR framework called TriOCR:** We designed a one-of-a-kind multi-engine pipeline that smartly combines outputs from three different OCR systems. The heart of this framework is its advanced alignment algorithms and a bespoke confidence-scoring system that uses the strengths of each engine to create a strong consensus and add an important quality assurance layer.

4. **Built and Used a Legal Lexicon for a Specific Field:** We constructed a whole Greek legal lexicon from a large collection of real documents because we knew that generic OCR doesn't work well with specialist language. We added this lexicon to a post-processing module to fix specific errors in legal terms, which greatly improved the overall correctness of the recognized text.

5. **Confirmed a Secure, On-Premise Architectural Model:** The whole pipeline was built and tested as a lightweight system that might work well in a resource-limited, on-premise setting. This shows a safe and useful approach to process very private legal papers without using outside cloud services.

1.2 Organization of the volume

Here is the outline for this thesis. Chapter 2 addresses the history and theory behind Optical Character Recognition non-technically. It uncovers the way the technology developed and determines the most widespread uses for it. Chapter 3 provides a detailed comparison of the best OCR technology currently available on the market. That includes analysis of the best open- and commercial-platforms, the most advanced deep learning models, and techniques for speeding up the process for documents.

Chapter 4 outlines the research problem based on the above explicitly. It elaborates on the special problems when dealing with legal documents step-wise and delineates the functional and non-functional requirements for the system under review. Chapter 5 tells you how the multi-engine, lightweight OCR pipeline proposed was developed and created. The chapter elaborates on the module architecture in tremendous detail, including the preprocessing, the semantic segmentation, the multi-engine coordination, and the post-processing stages.

Chapter 6 is about putting the system through its paces in the actual world. It describes the method, the carefully chosen set of Greek legal documents, and a detailed comparison of the proposed pipeline to the best commercial and open-source OCR solutions. Chapter 7 concludes the thesis by summarizing the primary findings, discussing the principal contributions and their implications, acknowledging the limitations of the current study, and proposing avenues for further research.

Chapter 2

Background

2.1 Optical Character Recognition (OCR)

Optical Character Recognition (OCR) is a field of study and technology that focuses on automatically recognizing and converting printed, typed, or handwritten text from physical or digital images into machine-readable, searchable, and editable data. It enables the transformation of non-digital documents, including scanned paper files, PDFs, and pictures containing text, into a format suitable for system processing, indexing, and computer analysis.

OCR essentially combines computer vision, pattern recognition, and natural language processing fundamentals. It was significantly dependent on machine learning and artificial intelligence, especially deep learning architecture like Recurrent Neural Networks (RNNs) for sequence text recognition and Convolutional Neural Networks (CNNs) for text localization. With such advancements, OCR system's performance and accuracy have become significantly higher regardless of the circumstances such as the use of non-standard or irregular fonts, poor resolution of the images, or multi-lingual input.

OCR finds universal usage across various industries. It finds intensive usage within banks for the automatic processing of checks, within archives for the digital conversion of ancient manuscripts, within educational organizations for converting hand-written material, and within government agencies for automating papers. Perhaps the most revolutionary sector where OCR has seen usage is the sector of law. Digitization of law documents—contract papers, case papers, court documents—has become increasingly pervasive as firms and organizations seek greater accessibility, search capability, and storage efficiency for papers. OCR finds a central place within the process since it can be utilized for converting paper-

based legal resources into structured digital material that can be retrieved immediately and subjected to analysis.

In addition, OCR continues to be a major target for research, especially for breaking the problem for handwritten documents, for low-resource languages, and for scene text recognition. As the size of information continues to grow and the demand for digital accessibility continues to grow, OCR continues to be a central technology for bridging the gap between digital and analog documents.

2.2 OCR History

To understand what OCR technology can do now, we need to look at how it has changed over time, from simple signal-based systems like Morse code and basic pattern recognition algorithms to the complex, AI-driven neural network models that are used today.

2.2.1 Setting the Stage (1914–1950s)

In the early 1900s, the first advances toward Optical Character Recognition (OCR) were made. In 1914, physicist Emanuel Goldberg invented a mechanism that could read letters and turn them into telegraph code. People know that this machine was one of the first to exploit OCR-like technology in a useful way [1, 2].

Goldberg kept conducting essential work by creating the ”Statistical Machine” in 1931. This electromechanical device was made to read optical codes from microfilm archives. People think this discovery is the first step toward modern search engines and a big step toward the idea of systems that can interpret machines [2].

ADavid H. Shepard, a cryptanalyst for the Armed Forces Security Agency (now the National Security Agency), built a machine called ”Gismo” in the 1950s. This was a big step forward. In 1951, Harvey Cook Jr. and I made ”Gismo.” It could read and write all 26 letters of the English alphabet as they were written on a standard typewriter. This was a big step forward in technology for recognizing letters [3]. In 1952, Shepard started the Intelligent Machines Research Corporation (IMR) so he could keep producing and selling OCR machines. A lot of people were interested in his new work, and in 1959, he showed off his creation on the American TV show I’ve Got a Secret as a machine that could read and write [3, 1].

These early changes were very crucial for the future advancement of OCR technology.

According to Wang [1], people in the 1950s and 1960s tried to make computers work less hard and allow robots comprehend conventional printed characters, especially English ones. These early mechanical and optical devices weren't very useful because they had small character sets and stringent guidelines for how to type. But over time, they would turn into the robust, learning-based OCR algorithms that are presently employed to read text.

2.2.2 Commercialization in the Early Years (1950s–1960s)

In the middle of the 20th century, Optical Character Recognition (OCR) went from being a theoretical experiment to a genuine technology. In 1954, Reader's Digest became the first company to deploy an OCR technology to transcribe typewritten sales reports into punched cards. This was a big step forward. Most people think that this adoption was a turning point that sparked further interest in automating document handling in enterprises [3]. At the same time, academics were trying out ways to turn two-dimensional character data into simpler, machine-readable forms. These methods were cheap to run but didn't work very well for recognizing characters [1].

In 1959, a few years later, IBM released the IBM 1287, the first scanner that could read handwritten numbers that was sold to the public. This new idea showed that OCR was moving toward processing less structured inputs, which was a sign of future improvements in handwriting recognition [1]. At the same time, there was a growing interest in character recognition systems based on optics and electronics in the academic world. Wang [1] says that early research used peephole and vertical scanning techniques to make input patterns easier to understand and help with early template-based matching.

The year was the 1960s when Ray Kurzweil and a few new folks started experimenting on how to discern patterns. He made a computer program that learned classical music and reproduced the patterns composers created when writing music when he was a teenager. It was a gargantuan step on the way to his future OCR work. He was to go on and develop OCRs for the blind later on in the 1970s, but his earlier work shows just how OCR evolved through the process of error and experiment iterated numerous times [3]. These early attempts, which ranged from business use to university research, laid the conceptual and technical framework for stronger OCR systems. Later, these technologies would gain from the addition of statistical approaches, machine learning, and finally, deep learning models.

2.2.3 Growth and New Ideas (1970s–1980s)

OCR technology was markedly advanced in the 1970s and the 1980s. Products for use with a limited number of fonts gave way in the years to products that could differentiate a larger number of fonts. Ray Kurzweil started the business of Kurzweil Computer Products, Inc. in 1974 and created the first omni-font OCR system, which was able to read text within almost all standard typefaces. Arguably the biggest disadvantage on the part of earlier OCR programs was the fact that they could only read certain types of fonts or formats. This fix solved that problem [3].

Kurzweil's omni-font OCR was a key step in making text processing systems that work for everyone. It was eventually added to assistive technologies like reading machines for people who can't see [4, 5]. In 1980, Xerox bought Kurzweil's company and changed its name to ScanSoft. The corporation continued to work on OCR technology for application in other businesses [3].

During the 1980s, OCR became quite popular in schools and libraries. Libraries and institutions started using OCR technologies to turn printed books, newspapers, and historical documents into digital files. This made it possible to make searchable digital catalogs and text repositories. This approach greatly improved the ability to access, save, and find information in schools and government offices [4, 1].

There was also a rising interest in using pattern recognition and optical approaches for OCR during this time. This set the stage for deeper integration with machine learning techniques in the years that followed [1, 5].

2.2.4 Adoption by the general public and integration with other software (1990s to early 2000s)

In the 1990s and early 2000s, Optical Character Recognition (OCR) became very popular. This happened around the same time when personal computers and digital processes became more common. Caere Corporation's release of OmniPage was one of the most important business events of this time. ScanSoft eventually bought Caere, and then Nuance Communications merged with ScanSoft. OmniPage started out in the late 1980s, but it truly took off in the 1990s as personal computers became more prevalent. During this time, companies and groups switched from managing documents on paper to managing them digitally. This

increased the need for OCR software that could run effectively on consumer-grade hardware [3, 2, 1].

In 1993, ABBYY FineReader was released. It quickly became an extremely accurate OCR tool that could read text in many different languages. Its launch helped OCR programs spread all across the world, especially in locations where people speak more than one language and do business with individuals from different nations [5, 1]. These tools were not like prior OCR technologies, which were usually stand-alone, hardware-based systems that needed to be set up in a certain way. OCR was added to other programs, such as scanner drivers and PDF processing tools. This made it a normal aspect of digitizing documents in the late 1990s and early 2000s [3, 1]. This period also marked the inception of interdisciplinary research that began to influence user-centric OCR systems. For instance, Cynthia Breazeal's work in robotics and machine learning helped define the ideas that went into AI-enhanced OCR encounters. On the other hand, Isabel Meirelles' work on information visualization and human-centered design changed how people formatted, understood, and moved through OCR outputs [6, 4].

These changes made OCR a more user-friendly technology that was employed by a lot of people instead of only businesses. This made it easier for AI frameworks to use it more in the future.

2.2.5 Cloud Integration and Deep Learning (2010s–2020s)

The decades since the 2020s and the 2010s have been the most critical decades for Optical Character Recognition (OCR) due to the developments made in AI, deep learning, and cloud technology. Google put out Tesseract version 4.0 in the year 2018. It was a tremendous upgrade on the existing open-source OCR technology available. It was a giant leap forward. Tesseract version 4.0, instead of using rule-based or outdated statistical models, uses Long Short-Term Memory (LSTM) networks, a kind of Recurrent Neural Network (RNN). It made it much more adept at distinguishing handwriting, multi-line layouts, and characters present in images which are corrupted or noisy [4, 1, 5].

By using deep learning, this application converted OCR into a data business rather than a business based on rules, and it made it far more adept at handling documents for a variety of languages and formats. Research today validates that the neural techniques reliably outperform the older OCR technology in accuracy and adaptability, particularly for handling

advanced visual input under natural conditions [1, 6].

At the same time, the addition of OCR to cloud-based services like Google Drive, Microsoft OneDrive, and enterprise-level AI solutions was another important step forward. These solutions let OCR programs run in real time on any platform, which makes high-performance text recognition available to a lot of people through web services and mobile devices [3].

The business organizations have also seen a shift in the manner they conduct business since they become more and more dependent on AI-based OCR. Ricardo Amper began Incode Technologies in the year 2015 as part of a grander vision for the expansion of OCR capabilities to biometric and identity verification programs. Such innovations reveal the truth that OCR can no longer be categorized as a text-reader; it is currently a deep component within autonomous-decision-making applications and receives intelligent information [4, 5].

2.3 Methodologies in OCR

Modern Optical Character Recognition (OCR) systems are generally structured around two principal methodological approaches: stepwise and integrated. These methodologies define how text is detected and recognized within images. In stepwise approaches, detection and recognition are treated as distinct, sequential processes—text is first located, then segmented, and finally recognized using a feed-forward pipeline. Most of the time, the method involves a coarse-to-fine process, where the initial candidates are slowly winnowed and confirmed. Integrated methods, on the other hand, seek to perform the two tasks simultaneously: find and recognize. They communicate information between the modules and usually make use of collaborative optimization methods. The products of character categorization are shared throughout the system, giving a common system which makes recognition faster and more accurate [3].

2.3.1 Text Detection

Text detection is a basic part of OCR systems. It involves finding text areas in an image before recognizing characters. Detection methods have changed a lot over the years. They have gone from heuristic and hand-crafted systems to complex architectures based on deep learning. In general, text identification methods can be split into two groups: classical

methods and neural network-based methods, especially those that use Convolutional Neural Networks (CNNs).

Traditional Detection Methods

Historically when deep learning was less prominent, OCR applications deployed a variety of outdated techniques for object discovery such as region-based and connected component (CC) algorithms [3]. A multiscale sliding window was applied by region-based methods to identify the best text regions within a photograph most of the time. The regions were then filtered using statistics like edges, histograms of gradient or intensity. Two popular types of classifiers that were utilized for the discrimination between text and non-text were Adaboost and Support Vector Machines (SVMs) [3]. These algorithms had high recall rates, but they were expensive to deploy and often generated false positives for the reason that they applied exhaustive search methods and features designed manually [4, 5].

On the contrary, CC-based methods cleared areas of images which were the same based on visual features which were the same, such as color, texture, or strength of gradient. It was successful as the text features were prone to appearing the same. Two well-known methods include Maximally Stable Extremal Regions (MSER) and Stroke Width Transform (SWT). These work well irrespective of what the scale, angle, or illumination conditions are. MSER, for instance, detects extremal regions within images which remain the same irrespective of the applied affine transformations. It works best when the intensity values are stable, and therefore it works best when it detects text within a scene [3].

The older text detecting algorithms failed when it came to layouts that could not be easily parsed, dirty sceneries, and non-standard fonts. They also struggled with real-time execution since they needed a decent amount of processor power [1, 6].

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of deep learning model that can automatically and adaptively learn spatial hierarchies of characteristics from images that are input. LeCun et al. (1998) first came up with the idea for CNNs. They were based on biological visual systems and were made up of many layers, with convolutional layers, activation functions, and pooling processes extracting and refining picture features at higher levels of abstraction [4]. A standard CNN has a number of convolutional layers that use filters that can

be learned to find patterns in the input image, like edges, textures, and forms. The outputs go through non-linear activation functions. The Sigmoid function was used at first, then AlexNet (2012) switched to the ReLU function, which was much faster and better at training. Pooling layers downsample the feature maps as the network gets deeper. This lowers the number of dimensions and the cost of calculations while keeping important spatial information [4].

CNN Architectures

CNNs are the basis for AI-driven document processing that finds text. They use hierarchical feature extraction to find text in different types of documents. The following architectures are very important:

ResNet

He et al. (2016) [7], introduced ResNet, which uses residual connections to solve the vanishing gradient problem in deep networks. This lets topologies have hundreds of layers (e.g., ResNet-50, ResNet-101). These connections help the network learn residual functions by adding the input of a layer to its output. This makes it easier to train deeper models. ResNet's depth makes it possible to extract strong features from text, which lets it find complicated patterns in difficult-to-read materials like historical or legal texts with noisy backgrounds [8, 9]. However, its processing complexity may restrict real-time use [10].

VGG

VGG was made by Simonyan and Zisserman in 2014 [11]. It is recognized for its simple and consistent architecture, which uses small 3×3 convolutional filters layered in deep layers (e.g., VGG-16, VGG-19). This architecture makes it easy to extract features while keeping the computer's workload minimal, which makes it great for organized documents like forms or printed reports [12]. However, it may not work as well on assignments with text or handwriting that changes a lot because it doesn't have enough different features [13].

EfficientNet

Tan and Le (2019) [14]suggested that EfficientNet uses compound scaling to balance depth, width, and resolution, which makes it more accurate with fewer parameters. Because it works so well, it's great for finding text in documents that are written in more than one

language and have different fonts, sizes, and orientations [15, 16]. EfficientNet's lightweight design makes it possible to use it on embedded devices, which makes it possible to do real-time OCR for mobile apps while keeping the best accuracy [3].

CNNs are great at finding complicated visual patterns, including character strokes, font changes, and background noise, that are hard to model with hand-crafted features. They don't need manually established rules because they can learn straight from raw image data. This makes them very good at both scene text identification and structured document analysis. They also support end-to-end training, where they learn to find and recognize text sections with high accuracy in different illumination, scale, and orientation settings [4].

CNN-Based Localization Techniques

Region Proposal Networks (RPNs): Two-step models such as CTPN and Faster R-CNN use RPNs. They make region proposals through the analysis of the feature maps and then transmit the region for classification and regression on the bounding boxes. The approach works well for high complexity layouts, such as technical drawings or contracts, but for real-time applications, it does not fit well given the expensive cost of computation [5, 17, 6].

Single-Shot Detection Methods: Unlike RPNs, single-shot detectors perform localization and classification simultaneously in a single forward run. Optimized for real-time applications like reading ID documents or structured forms, models like EAST and YOLOv8 are the best [18, 15]. Their accuracy can be a concern on crowded or busy text scenes compared to a two-stage approach [4].

2.3.2 Text Recognition in OCR

Text recognition translates the text regions within a photograph into computer-understandable strings, for instance, words or phrases. Previously, utilized as an image annotator, text recognition was best calibrated for simple, well-organized, and high-contrast inputs—typical for printed black-and-white books.

The newer OCR tools, however, have to assume more stringent scenarios, for example, natural scene photography where text is found within diverse fonts, lighting, distortion, or noise within the foreground [3].

In the OCR process, text recognition is usually performed after text detection, interpreting the contents within detected text bounding boxes. You can identify sequences, words, or char-

acters at different levels of detail. Character recognition involves the recognition of isolated characters with methods such as CNNs, but word-level recognition usually gets modeled as a sequence prediction problem, which requires learning contextual dependencies—such as Recurrent Neural Networks (RNNs) or mixed CNN-RNN models [3].

Traditional Methods for Text Recognition

Before the introduction of deep learning approaches, text recognition in OCR systems relied mainly on traditional machine learning pipelines and custom feature engineering. These methods usually used a multi-step process based on general object detection, which comprises region proposal, feature extraction, classification, and post-processing [4].

Recurrent Neural Networks in Text Recognition

Recurrent Neural Networks (RNNs) are a type of neural model that are developed for the purpose of dealing with ordered input and output sequences. The RNNs have recurrent links which enable them to carry information that was observed on earlier stages within the sequence and therefore can effectively model temporal dependencies between elements [4].

In text recognition, the RNNs are particularly well-qualified for handling sequences of characters such as lines of text or words. In OCR, when it's applied, the input is a sequence of visual features—most often extracted from an image using a CNN—and the output is a sequence of guesses for characters. The RNN, on every time step, produces a distribution over possible characters or a continuous code, depending on the task. The model learns through the optimization of a loss function (usual cross-entropy) which monitors how far away the sequence guesses for the characters are from the correct ones. It does so through backpropagation through time, which adjusts the parameters.

RNNs have remarkable advantages for OCR due to the maintenance of relationships between contexts for characters, a need for the detection of complex or confusing text—more so for scene text recognition, where the characters are deformed, noisy, or non-linearly arranged.

Convolutional Recurrent Neural Network (CRNN)

A Convolutional Recurrent Neural Network (CRNN) is a hybrid deep learning architecture that merges CNNs for visual feature extraction with RNNs for sequence modeling.

CRNNs are made to read text as a series of spatial properties, so they can recognize whole words or lines of text without having to break them up into individual characters.

The CNN part of the system picks up on hierarchical patterns and local structures in the image, like edges, strokes, and character shapes. The RNN part models the temporal or sequential dependencies between characters, which lets the system read text in a logical order, like from left to right or from top to bottom. This combination tackles the constraints of earlier OCR systems by integrating spatial and sequential learning into an end-to-end trainable architecture.

CRNNs have significantly improved recognition accuracy and robustness in complex scenarios involving distorted, curved, or noisy text, commonly found in natural scene images or historical documents. They eliminate the need for manual character segmentation and can accommodate variable-length input, making them adaptable and efficient. As a result, CRNNs have become a standard architecture in modern OCR systems and are widely used in applications such as license plate recognition, document digitization, and mobile text reading [4].

The combination of CNNs and RNNs in CRNNs shows the bigger change in pattern recognition, from hand-made features to deep, trainable models. This is a big step ahead in the history of OCR.

Attention Mechanism in OCR

The attention mechanism is a neural network component meant to improve performance when processing sequences, especially ones with changing lengths or intricate dependencies. Attention was first used for neural machine translation in 2014 [4]. It was inspired by how people can selectively focus on important parts of visual or linguistic input.

In OCR, attention mechanisms are typically employed in models that need to recognize unsegmented or distorted text sequences. When used with CNN-RNN hybrids, attention allows the decoder to dynamically focus on different regions of the visual feature map while predicting each character or word. This helps the model connect input areas to output symbols, even when the layout of the space is strange or noisy [4].

2.4 OCR Phases

There are numerous elements that work within a complete Optical Character Recognition (OCR) system for the purpose of reading text effectively and precisely. These stages are ubiquitous within those stages: pre-processing, segmentation, normalization, feature extraction, classification, and post-processing. These constitute the composition of a typical OCR pipeline, spanning the preprocessing of the original image through the generation of searchable and actionable text.

Each step handles one particular problem: the pre-processing gets rid of noise and cleans up the image; segmentation splits words or characters; normalization normalizes forms and sizes; feature extraction identifies surprising patterns; and classification gives names to words or characters. Afterward, the post-processing uses language models or correction programs for refining such results. [1].

2.4.1 Pre-processing Phase in OCR

All OCR systems begin with the pre-processing step. The goal here is for the input images to be good and set for the accurate identification of characters. The general goal here is the elimination of noise and irrelevant information and keeping essential features for segmentation, feature extraction, and classification [19].

Image-based OCR programs that operate on color, gray-scale, or binary images need a good deal of preprocessing. Most OCR programs render the color images into black and white or binary so the mathematics can be reduced and the process can proceed faster. It takes a vast amount of computer power for the computer to operate on the color images. The process is vitally important for getting rid of artifacts and inconsistencies and for rendering recognition more consistent.

Some things that happen a lot during pre-processing are:

- Binarization: transforms photographs from grayscale to binary so that you can read the text and the background independently.
- Noise Reduction: Removes excess pixels that are there because of scanning artifacts or noise in the area.
- Skew Correction: Fixes the rotation of a document so that the text lines are straight.

- Morphological Operations: Get rid of faults or fill in gaps to make characters look better or change their shapes.
- Thresholding: This makes it easier to differentiate text from the backdrop by using pixel intensity thresholds.
- Thinning and Skeletonization: This makes the strokes of characters only one pixel wide, which makes it easier to look at shapes.

These steps make sure that the input image is clean, even, and constant, which makes the OCR system work much better [19].

2.4.2 Segmentation Phase in OCR

The segmentation step plays a huge part in the OCR pipeline. It splits the pre-processed image into sections usable for the recognition step, like lines of text, individual words, and individual letters. Segmentation accurately is extremely important as it directly affects the performance of the recognition step. A Detailed Analysis of Optical Character Recognition Technology reveals that segmentation most frequently takes place within a hierarchy, starting with the text line isolation, the word level, and finally the character level. The approach becomes especially problematic when layouts are skewed, overlapping characters are a result, or noise lingers from previous stages.

To separate characters or symbols and retain the character of the structure, segmentation efficiently embodies methods like projection profiles, component linking analysis, and edge detection. Segmentation errors often propagate into recognition and lead to the misclassification or illegibility of the text, most prominently on degraded or handwriting documents. Segmentation therefore not only enables the image structure to deteriorate but connects the low-level image processing and the high-level pattern recognition for OCR applications.

2.4.3 Feature Extraction Phase in OCR

The feature extraction phase is an important part of the OCR workflow. It extracts the most prominent attributes of separated characters and translates them into organized forms called feature vectors. It is what the classifiers use when they want to identify what a character or word is. Feature extraction, as per A Detailed Analysis of Optical Character Recognition Technology, is successful when it makes classification a simpler matter by showing

clear characteristics which unmistakably distinguish a character class from another. There are a lot of different methods that have been suggested in the literature, including directional chain code features, zoning, end-point detection, and diagonal-based extraction. Each of these works best with a distinct set of characters and writing style.

There are two main types of feature extraction techniques: statistical and structural. Statistical features are calculated using techniques like zoning, moments, projection histograms, and Fourier transformations. They look at how pixel values are spread out. Because they depend on averaged pixel values in smaller parts of the character image, these are often called ”global features.” The structural features, however, originate from the geometric and topological characteristics of the characters, for example, loops, ends, and convexities. These features indicate how the character is built up and are most helpful for discrimination among handwriting or stylized scripts. The system works best when you choose features such that they are unique and robust enough for handling changes of style, scale, and noise.

2.4.4 Classification Phase

The classification phase in OCR is what puts the features that have been taken out into the right character or word class. It is an important part of pattern recognition since it uses several algorithmic methods to figure out what segmented symbols are. We talk about the most important ways to classify things in OCR below.

Template Matching

Template matching was among the original and most straightforward OCR techniques. It compares images of input characters with a set of stored templates that correspond to known characters. Techniques like pixel-wise or curvature comparison are utilized for determining matches. The technique works well when dealing with clear, printed text and typical font face, but it’s quite sensitive when it comes to noise, distortion, and non-standard font. It works poorly when the input is handwritten or scripts are non-standard [19, 1].

Statistical Techniques

Using probabilistic models to assign characters based on how likely it is that certain traits are present is what statistical categorization does. Some common methods are Nearest Neighbor (NN), Bayes classifiers, Hidden Markov Models (HMMs), clustering analysis, and fuzzy

logic. These models learn from how the features are distributed and can change to fit different shapes and styles of writing. But they often need well-chosen features and big training datasets [19].

Neural Networks

Feedforward, CNN, and RNN-based ANNs have improved OCR significantly. ANNs (artificial neural networks) are learned to transform input into output classes by changing the weights within them. It's modeled after the functioning of biological neurons. CNNs can learn spatial features more effectively, and RNNs can learn sequences more effectively. Hybrid models such as CRNNs are increasingly being adopted in order to manage rich and unsegmented scripts effectively. Most modern OCRs attain accuracy rates greater than 95% using the above models [1].

Support Vector Machines

SVMs are robust supervised learning methods that can be utilized for multi-class and binary classification. They work on determining the hyperplane with the most freedom that optimally separates classes. If the data can be separated and correctly labeled, the SVMs perform best on high-dimensional spaces. Kernel-based expansions, such as Kernel PCA or KFDA, enable the SVMs to accommodate non-straight line boundary decisions, which makes them greatly more flexible [1].

Combination Classifiers

Combination or ensemble classifiers use the best parts of several classification algorithms to make recognition more accurate and reliable. OCR systems can make up for the drawbacks of each method by mixing several models, including Neural Networks and k-Nearest Neighbor (k-NN). This method works best in real-life situations where the inputs are noisy, the fonts are different, or the layouts aren't conventional [19].

2.4.5 Post-processing Phase in OCR

The last and most crucial step of the OCR process workflow is post-processing. It's purpose is to improve the recognition system's accuracy by fixing errors still remaining. Catego-

rization and preprocessing look at images and letters, but using language and context, post-processing checks and corrects the text detected [19]. We can often make educated guesses what poor or skewed writing says based on the situation. Post-processing tries the same thing using algorithms. A common way of fixing things here is the use of a dictionary. It does it by comparing the OCR output with a lexicon and finding and fixing words that are spelled or detected erroneously. It's similar to a spell checker since it makes suggestions or changes the text automatically based on what it's comfortable with. And context-sensitive models might have a use for grammar rules or statistical language models and inspect word sequences even deeper, especially for longer pieces of text. The step of post-processing can also consist of putting the text recognized into structured digital outputs like PDFs or spreadsheets, so it can easily be exploited and searched later on. That's on top of fixing the language. The step of post-processing does more than just improve recognition accuracy; it also makes the final output relevant, searchable, and semantically meaningful [19].

Chapter 3

State of the art

Although OCR performance is impressive with accuracy over 99% for high-quality machine-printed documents, on real-world applications such as handwritten text recognition, degraded document quality, multilingual processing, multistage recognition systems, and complex document layout analysis, OCR is far from being solved. In this chapter, we aim at presenting an overview of the existing works in the area of OCR as well as in the realization of OCR engines that exploit the potential of deep learning in order to achieve high-level recognition systems.

We look at Tesseract OCR, an established open-source engine, as well as newer deep learning-based solutions including PaddleOCR and EasyOCR, and cloud-based offerings from leading technology companies including Amazon Textract and Google Cloud Vision API. Discussion is not only limited to single tools but also includes state-of-the-art methodological advancements like transformer-based architectures, Vision Transformers as well as "hardware acceleration" methods that are redefining the performance boundaries of OCR systems.

In doing so, we establish the technological basis and competitive environment that motivate the design and implementation of our AI-driven document processing acceleration system that we propose, and thereby define opportunities for innovation, and benchmark standards modern OCR technologies can be measured against / must be improved.

3.1 Modern OCR Frameworks and Libraries

The contemporary OCR ecosystem is dominated by a diverse array of frameworks and libraries, each offering distinct advantages in terms of accuracy, performance, language support, and deployment flexibility, requiring careful evaluation to understand their capabilities and limitations in modern document processing applications.

3.1.1 Tesseract OCR

Tesseract OCR is a popular, free, and open-source OCR engine developed first by Hewlett-Packard and, since 2006, maintained by Google, that has become one of the best OCR tools used for academic as well as other research purposes. With its broad language coverage and post-4.0 state-of-the-art LSTM-based recognition engine, Tesseract is among the most accurate open-source OCR engines on the market.

The advances between Tesseract 3.0 and versions 4.0 and 5.0 have shown dramatic increases in read accuracy and recognition speed. The big step forward was a new LSTM-based recognition engine with Tesseract 4.0, which increased the quality considerably, especially for complex scripts and languages. This integration of deep learning has enhanced Tesseract in processing sequential dependencies of text better than previous solutions employing only CNNs, and it achieved high precision in recognition of English handwritten text [20, 21].

Recent benchmarks also show very good performance improvements with Tesseract when used in conjunction with optimized preprocessing and hardware acceleration. The average latency is 48 msec per image for the CPU-based serial processing, and with the GPU-based parallel processing and streaming capability, we are able to further reduce it to a modest latency of 0.825 ms, a speedup factor of 58.6x faster. This is an impressive boost in performance and was one of the reasons why we chose to port Tesseract to the GPU [22].

The performance of Tesseract is highly sensitive to both preprocessing methods and configuration options. The highest gains in accuracy are achieved through specific preprocessing methods such as thresholding (e.g., Otsu, Sauvola), binarization, denoising by median filtering and morphological operations, deskewing and rotation correction, and adequate scaling. These preprocessing procedures also increase the accuracy recognition on different datasets and image types [20, 22].

Setup options also play an equally crucial part in performance tuning of Tesseract. Im-

portant settings include selecting the right language with the -l parameter, the proper Page Segmentation Mode (PSM) for the type/layout of the document, using the LSTM-based OCR Engine Mode (OEM 1 or 2) for modern documents, and applying dictionary-based post-processing for complex scripts. The combination of powerful preprocessing and customized configuration is key to obtaining the highest accuracy for a wide variety of document types [23, 13].

It is, however, limited in a variety of use-cases in many challenging scenarios, despite its wide use and constant updates. For handwritten text, Tesseract performs quite poorly (for cursive or stylized handwriting, even preprocessing and training on large, varied datasets will get you nowhere near acceptable results!). Recognition rates of handwritten documents are still, in general, below those obtained with specialized or commercial OCR engines [24, 25].

The multi-column layout issue in Tesseract is another difficult problem of layout analysis. The engine struggles at times to accurately segment complex document structures, sometimes interpreting reading-order novel, questionable column orders, or stale content. The comparison also illustrates that Tesseract, as an OCR API based frequently on book fonts for training purposes, is less successful as a generic work and process output content parser for documents with complex layouts like Amazon Textract [24].

Tesseract's biggest weakness is probably its handling of non-Latin scripts. While the engine can handle a wide range of languages, script-specific complexities, complex text shaping, and the inability to develop good language models have contributed to poor performance dealing with scripts such as Thai, Persian, Cyrillic, or multilingual script text. This is because, in order to obtain high accuracy, we often need to custom train and fine-tune for the best-fitted large high-quality datasets for each script, and even so, error rates tend to stay higher than those of Latin scripts [20, 13, 26].

Though the LSTM-based architecture of Tesseract represents a leap forward compared to traditional template-matching approaches, recent studies have demonstrated that newer deep learning models can obtain better performance than Tesseract in challenging scenarios. Region-based CNNs (R-CNNs) and other hybrid models have particularly better performance, especially for text printed with a different technique or on a curved plate, which is almost impossible for Tesseract to meet the requirement. For example, some R-CNN techniques perform 91.1 compared to Tesseract 38.3 in a specific and challenging case of several print methods [27].

Despite these shortcomings, Tesseract is a powerful OCR tool that can be used effectively for many tasks, especially when using it with other libraries (like OpenCV for preprocessing) and optimizing the configuration to the task at hand. Its open-source nature, the good documentation, and the active support of its community keep attracting both researchers and developers who are looking for a stable and customizable base for their OCR-related work.

3.1.2 PaddleOCR

PaddleOCR is an advanced OCR toolkit rapidly developed using PaddlePaddle, Baidu's Parallel Distributed Deep Learning, which, based on the innovations of PP-OCR series solutions, addresses the demands of multilingual text recognition with lightweight, and extensive deep learning models for both high accuracy and deployment efficiency in various computing resources.

The PP-OCR series has made a great stride advancing the development of OCR technology by integrating effective and efficient architectures together with the current state-of-the-art for text detection and recognition. The architecture includes a number of important design innovations such as lightweight networks like SVTR_LCNet for text recognition that strikes the balance between speed and accuracy, hence, suitable for real-time and embedded systems. The text detection module adopts the Differentiable Binarization for accurate text region localization and presents efficient performance in terms of text region localization on complex document images [28, 29].

The strong preprocessing pipeline of PaddleOCR, ranging from grayscale as well as bilateral filter, to rich data augmentation such as random rotation, flip, contrast and brightness adjustment. These preprocessing methods are very useful to increase the generalization and recognition accuracy of the model, in noisy and heterogeneous environments [28].

The great advantage of PaddleOCR is its multi-language support. The system reaches high f-measure up to 93.89% at multilingual and hybrid texts annotation, proving wide adaptability in dealing with several scripts and technical lexicons. This is realized by the incorporation of CRNN and SVTR models that support custom vocabulary and domain-specific training methods [30].

PaddleOCR utilizes advanced training methods to achieve good performance with a wide range of languages. The cross-language knowledge transfer training approach will use data from high-resource languages to obtain improvements in low-resource languages, thus achiev-

ing efficient transfer of knowledge across linguistic divides. The model is also trained with weighted updates and neighborhood distribution augmentation, as well as ensemble adversarial training, which helps the model to be robust to adversarial and noisy inputs, and improve the generalization power respectively [31].

Data augmentation is very important to the multilingual capability of PaddleOCR. For this, the framework utilizes multilingual data augmentation that augments data by generating synthetic or translated samples in target languages, which can systematically fill in the gaps where annotations are limited (also acts as a support to zero-shot and few-shot learning). Other approaches involve Perturbation Embedding, which adds controlled noise to enforce training examples to be less sensitive to real-world input variation, and chain-of-thought prompting for challenging semantic tasks [32, 31].

The small models of PaddleOCR are tailor-made to solve computationary limitations in mobile and edge. The system has a number of advantages for resource-constrained deployments, including significantly smaller model sizes compared to legacy OCR frameworks, making them suitable for deployment on devices with low storage and memory. Performance Benchmarks.

Performance of PaddleOCR is 40% faster than some other contrastive models, ensuring that the user experience is smooth on mobile and edge devices [16].

Moreover, the proposed lightweight design exhibits low computational and memory overheads thanks to small model size; hence leading to reduced CPU/GPU utilization and less battery overheads, being crucial for mobile and battery-based gadgets. Of important note, PaddleOCR exhibits high levels of accuracy with minimal drop in its performance once ported to edge deployment runtime primitives such as TensorFlow Lite, thus ensuring that it is highly deployable for realistic, resource-constrained environments [29].

Compared with other open-source OCR frameworks, PaddleOCR has an advantage in the model size, the inference speed, and the accuracy. Performance comparison tables show that PaddleOCR models are approximately 4x the size of ultra-lightweight solutions such as MULDT (3.6 MB vs 890 KB), and still much smaller than conventional frameworks like CRAFT which can range up to 89x larger than the most minimalistic ones. In terms of speed, PaddleOCR is up to 40% slower than the highest-performing lightweight models of which we are aware, however, it still competes in speed with traditional larger-size frameworks [16].

The accuracy performance of PaddleOCR is competing, this achieves a recall rate of

0.792 on the MIDV-2019 dataset, which surpasses CRAFT's 0.718 while trailing behind the ultra-lightweight MULDT at 0.826. This performance profile situates the PaddleOCR as a good candidate for practical applications that require a balanced trade-off between accuracy, efficiency, and deployment decoupling [16].

PaddleOCR aims for practical deployment on various devices and various scenarios. The efficiency of the framework made it especially suitable for on-device real-time document scanning in smartphones, embedded scanner devices, AR glasses, and other resource-limited systems. Ease of use and deployment are hallmarks of PaddleOCR and the addition of edge-friendly deployment ensures that PaddleOCR's state-of-the-art text recognition system can be used when commonly required and when network connectivity may not be possible, it's still there as a feature-rich and functionally complete document analysis engine.

3.1.3 EasyOCR

EasyOCR is a Java-based OCR library that leverages modern deep learning techniques to offer an efficient, accessible information extraction tool for written text, especially focused on ease-of-use, multilingual support, as well as practical use in various complex document and scene text scenarios.

EasyOCR uses an advanced two-stage model which consists of the most advanced methods for text detection and recognition. For the text detection, the system mainly relies on EAST (Efficient and Accurate Scene Text Detector) algorithm which is notable for its speed and precision. By detecting text in arbitrary orientations, this is particularly well suited for complex layouts and real-world scene texts. The EAST detector is preferred in EasyOCR for its capability to deal with different text orientations in addition to its performance-efficiency tradeoff [17, 33].

The OCR module is augmented with an attention mechanism and is based on a Convolutional Recurrent Neural Network (CRNN) architecture. This complex architecture includes convolutional layers for feature representation, recurrent layers, typically based on LSTM, for sequence representation, and a transcription layer, usually using CTC loss, that maps image features into text sequences. The CRNN structure can be further refined with a soft attention mechanism which allows the model to concentrate on the informative parts of the input image, leading to a significant increase in recognition accuracy, especially in challenging or cluttered backgrounds [34].

The recurrent part of the CRNN involves LSTM layers with tailored upgrades aimed at improving both performance and resiliency in complicated industrial and industrial-like environments. This architecture allows EasyOCR to effectively model sequence dependencies in text, it is especially good at recognizing continuation strings and keeping context between characters [34].

The multilingual support of EasyOCR is one of its major advantages. It comes with pre-trained models for many languages and provides an easy to use API for multilingual OCR tasks. The software permits concurrent use of multiple target languages in a single processing scenario, which means that one can recognize the content of interest in several languages within the same image or document. This can be done for the problem at hand by loading and using relevant pre-trained language models for detecting and recognizing text in any of the mentioned languages in a single processing step [35].

Multilingual recognition includes advanced character and word detection capabilities to facilitate handling of documents containing text in multiple languages by recognizing and processing their respective scripts and structural features. This procedure allows EasyOCR to use multi-lingual situations as are often presented in real-world documents, e.g., international business documents, multi-lingual signs, academic papers with multiple scripts [36].

EasyOCR support for language, although broad, pales in comparison to other OCR libraries e.g. Tesseract, especially for low-resource or uncommon languages. Moreover, variability may exist with regard to the populations, language combination, and (specific) complexity levels of the layout or script of the document to which recognition is to be applied and this needs to be taken into consideration when applying the system in specific multilingual scenarios [35].

The performance of EasyOCR is highly correlated with image quality in terms of accuracy and speed. The model also provides good quality of results, very quickly, in the case of high-quality images with clear symbol-background contrast and where the noise level is very small. On the other hand, the performance is significantly deteriorated when lowering the quality of the available images, and in such cases, the accuracy of the model may decrease and the inference time may increase [37].

Several image quality factors affect performance: Images with higher contrast, and lower levels of noise tend to result in higher accuracy, and create less ambiguity between similar characters (e.g. ’o’ and ’0’, ’i’ and ’1’). The degradation of blur and resolution greatly weaken

both the capabilities for character detection and recognition, and the detection results are even not available when the character sizes are reduced to around 40-30% of their original scales. In light of these challenges, EasyOCR exhibits similar performances with other OCRTs even under strong noise and poor contrast conditions, although state of the art will need further fine-tuning under highly degraded images [37].

Speed wise, EasyOCR should remain quick and efficient for bright and clear images, but processing time may increase as image quality degrades and when you add preprocessing steps to improve the input. The accuracy can be increased on poor-quality images with fine-tuning by seeing the binarization, sharpening and filtering techniques but will lead to increased computational complexity affecting the overall processing speed [38].

CRAFT (Character Region Awareness for Text detection) algorithm implemented by EasyOCR has impressive strengths in detecting character-level texts and is especially good at solving texts in different directions. Yet, relative performance analysis discloses a number of performance trade-offs for new detection techniques. CRAFT provides a fair accuracy with a recall of 0.718 on MIDV-2019 document dataset, which is underperformance PaddleOCR (0.792) and ultra-light summer models such as MULDT (0.826) [16].

From an efficiency standpoint, CRAFT is unsuitable for contemporary deployment due to its size and speed relative to the modern lightweight models. For example, MULDT is 89 times smaller and much faster than CRAFT. And CRAFT is much more suitable for embedded or real-time applications with low resources. Furthermore, CRAFT might fail in several instances of multiline text, as it often clusters bounding boxes within consecutive lines and yields erroneous texts under different formats of complex document layouts that necessitate further post-processing operations complicating deployment [15].

Experiments on the CRAFT dataset demonstrate that more recent models, as YOLOv7, YOLOv8, and Mask R-CNN, often achieve better accuracy and efficiency than CRAFT. These state-of-the-art methods can reach 97.5% weighted mean Average Precision (wmAP) with the benefit of faster processing and less post-processing operations [33, 18].

EasyOCR is designed with ease of use and ease of deployment in mind, it’s ideal for those who are developing and researching. Many OCR engines in the market have easily kicked off the ladder due to bad documentation or having too many configuration needs. An easy-to-use application programming interface (API) and good documentation regarding the framework also allow its extension in academic as well as commercial endeavors.

The EAST detection and feature attention-enhanced CRNN recognition could effectively detect and accurately recognize texts of various forms across programs, resulting in a robust performance on both structured document processing and unstructured scene text recognition. EasyOCR might not be the absolute top performer in terms of accuracy across the board, but its combination of convenience, simplicity of use and support for a wide variety of languages, make it a viable option for many real-world OCR use cases, where ease and speed of development and deployment are critical.

3.1.4 Amazon Textract

Amazon Textract is a fully managed machine learning service on AWS which automatically extracts text and data from scanned documents, automatically reflowing the content according to the understanding of how words are visually presented on the scanned page, as well as preserving the position of the words in the document, and it does not require rule-based templating.

Amazon Textract uses advanced machine learning algorithms to detect printed text and numbers in a wide variety of document types at even faster and more accurate rates, including text in virtually every printed or handwritten form, such as tabulated data, scanned documents, and tables. The machine learning models underneath are designed not only to detect characters but to learn the semantics of the document, enabling the service to preserve and extract the document structure such as tables, forms, or key-value pairs, hence converting unstructured data into structured types as JSON or CSV [39, 24].

The service is powered by Convolutional Neural Networks (CNNs) for image analysis and feature extraction, Recurrent Neural Networks (RNNs) and attention mechanisms for sequence modeling algorithms, and allows for context understanding via text analysis. Such models have been pre-trained on large-scale datasets that include various types of documents, fonts, languages, and layouts, and exhibit strong generalization performance in different practical settings. In comparison studies, Textract is shown to achieve accuracy levels using up to 95% for printed documents and 89.4% for handwriting datasets, outperforming other tools even for complex or tabular data where the structure of the document is maintained [40, 24].

Amazon Textract is based on the same proven, highly scalable AWS writing public cloud infrastructure, a containerized micro-services architecture, and it's designed for high-throughput, low-latency processing for real-time as well as batch document processing. Service-level par-

allelized task execution and intelligent queuing schemes are adopted for efficient scaling of large volumes of documents with minimal latency in processing and optimal resource utilization [39, 24].

The infrastructure can scale resources on-demand to support the workload request, whether for individual, small document processing or batch, large document production over thousands of documents. It is because AWS provides an auto-scaling feature that monitors job queues and usage statistics and automatically brings up more compute when needed and scales down when not needed to save costs.

One great thing about Amazon Textract is that it's really easy to use with the rest of the AWS Services. The service comes built-in with native integration with Amazon S3 for document storage, Amazon Kendra for semantic searching, AWS Lambda to build serverless processing workflows, and Amazon DynamoDB to host extracted metadata and structured data. This full-stack integration facilitates end-to-end document management and analysis workflows such as triggering document processing, maintaining results, and allowing complex search and analytics features [39, 41].

The service also offers powerful synchronous and asynchronous processing modes through its APIs. The synchronous processing is well-suited for real-time applications that need an immediate answer or as a simple use-case to provide quick results, and asynchronous processing works well with large documents or batch operations. The APIs feature clear, easy-to-use developer interfaces, and support multiple programming languages and frameworks, allowing them to be quickly and painlessly integrated into new or existing applications and workflows.

In addition to conventional OCR features, Amazon Textract also has document understanding to identify and extract whole documents, like forms, tables and key-value pairs with amazing accuracy. Form invocation, detection of form fields and corresponding values, extraction of tabular data, preserving of row and column relationships, and detection of locations of signatures, and states of checkboxes. Such expertise is particularly useful in handling business documents, e.g., invoices, contracts, financial statements, or regulatory forms.

It can handle multi-page documents to maintain document structure and reading order across pages for complex documents like annual reports, technical manuals, or legal documents. Furthermore, Textract is able to process documents with combined content types such as documents that have printed text along with handwritten notes.

Amazon Textract takes advantage of AWS's proven security and compliance capabilities and has also been tested against rigorous security standards, ensuring data is encrypted at rest and in transit, while also offering IAM-based access control and VPC endpoint support for secure access to Amazon Textract over a private network. The service is compatible with standards like HIPAA, SOC, PCI DSS, and GDPR, which makes it appropriate for handling sensitive documents in regulated industries such as healthcare, finance, and government [39].

Data privacy is enforced by the shared responsibility model delivered with AWS, the policy of which dictates that AWS secures the underlying infrastructure, and customers are responsible for how they use AWS services to store and retain their own data. Textract doesn't retain or use your intellectual property for model training unless asked to do so by the customer, providing them full control of their content.

Amazon Textract Pricing Amazon Textract follows a pay-per-use pricing structure, charging for the number of pages you process, so it is cost-effective if you need to process varying amounts of documents. There are also different pricing plans for specific lines of analysis with base text extraction analysis costing less than more advanced document analysis like forms and tables.

Performance is optimized by intelligent document preprocessing and adaptive model selection according to document characteristics, as well as by optimal utilization of resources on AWS's global infrastructure. The system is able to handle documents in any of the following formats: PDF, JPEG, PNG, TIFF, and process them with automatic format detection and content optimization for processing.

Amazon Textract delivers high accuracy and scalability, but organizations should weigh data residency requirements, reliance on the internet, and continued operating costs when comparing against other on-premises or cloud-based platforms. The application is especially useful for companies that are already using AWS and need a high level of scalability and integration with other cloud services.

3.1.5 Google Cloud Vision API

Google Cloud Vision API is a powerful and robust cloud-based computer vision service that belongs to Google Cloud Platform and can take advantage of numerous cutting-edge machine learning and deep learning technologies to detect, analyze, and understand images. Its OCR feature is the most used one for reading and extracting text from images, documents,

and PDF files in various languages and types.

The OCR model is part of Google Cloud Vision API, which is based on advanced neural network models, and works along with several deep learning instruments to continuously improve accuracy for a wide variety of documents and languages. The bottom of the system uses Convolutional Neural Networks (CNN) as its base to learn spatial features from the image. It can make the recognition of characters in a variety of fonts, sizes, and directions more robust. State-of-the-art CNN architectures such as GoogLeNet variants with global average pooling at the fully connected layers are employed to enhance the accuracy and XOR overfitting, especially for large-scale character sets [12, 42].

The API supports hybridization that consists of CNNs for feature extraction and RNNs—particularly LSTM networks—for sequence modeling. The CRNN model provides a powerful framework for recognizing whole text lines or entire words at once, but also for solving specific problems in text recognition that involve character sequence dependencies. The architecture uses Connectionist Temporal Classification (CTC) as a transcription layer to sound out text from the RNNs outputs, dealing effectively with sequence length variability without needing explicit character separation.

Google Cloud Vision API uses advanced training techniques that mainly rely on transfer learning – pre-trained models are either re-trained on new data or fine-tuned with domain-specific data. By using this technique, the system can be easily extended to support scripts or document types with little extra training data and can be quickly deployed for many languages and narrow domains [42].

It has been trained using strong data augmentation: rotations, zooms, shear, sharpening, and brightness/contrast transformation on training images. These augmentation techniques enhance the dataset diversity and the model's robustness to real-world variations in the quality of documents, lighting conditions, and types of image capture. Moreover, it uses a large-scale synthetic dataset generated by synthesizing the characters with different fonts and the backgrounds along with adding the real-world noise and conditions of the documents for improving the generalization of the model [42, 43].

Google Cloud Vision API shows good results for multiple languages and scripts but the accuracy ranges widely depending on the complexity of the script and writing style. Recent multilingual models that jointly learn script identification and text recognition tasks simultaneously have produced highly accurate results across English, French, Kannada, Urdu, and

Bangla languages and often outperform the systems for individual scripts [44].

The server-side API, though, still has difficulties with complex scripts like Arabic and Chinese because of varying character shapes, cursive connections, and position. Although ensemble models that combine CNNs and transformers have achieved strong accuracies of 89-98% in annotated handwritten Arabic text, errors still exist as a result of visual similarities between characters and ground-truth labeling difficulties [45].

Handwriting recognition suffers a lot from the variability of writing style, where large inter- and intra-class variances of handwriting forms often deteriorate the recognition performance. The accuracy of the API falls down severely under very poor or very irregular handwriting, and a study on dysgraphia indicates the accuracy rates of nearly 67.79% in case of letters and 52.36% for words [46].

While we don't offer a dedicated API for software testing, I would say Google's Cloud Vision API would be a good fit given its ability to detect the structure of a document. We can process each page as images and then extract the textual content which you could edit, save, or continue processing. The API allows for detection and extraction of text blocks, lines, and words, as well as offering positional information in the form of bounding boxes for each detected element, thus allowing rudimentary reconstruction of document structure such as separation of paragraphs, headers, and sections [46, 47, 48].

In cases of more complex documents, the API can be combined with object detection algorithms to find and locate salient objects (i.e., tables, logos, form fields) in a structured document, which would improve one's ability to extract information out of structured documents. This hybrid method allows for the unsupervised extraction of information from densely structured documents such as technical datasheets, where OCR is responsible for extracting the text and enabling other models to handle complex layouts and tables [49].

The API is commonly used in conjunction with other CV models or APIs to build scalable, do-everything workflows for document processing. Such workflows typically require a user to upload images or pages of PDF documents to cloud storage, process the images or pages to extract text using the Vision API, and stream the results back for doing further downstream tasks like data entry, editing, or information retrieval [50, 51].

To get the highest OCR accuracy using Google Cloud Vision API, one needs to carefully preprocess images and manage the workflow. Several important preprocessing approaches which considerably improve the recognition performance have been discovered through re-

search. Image sharpening and brightness adjustment work very well with documents with faded or low contrast text, and contrast enhancement is effective to improve the text-line-background separation for OCR engine [43].

Optimal resolution of the page image is critical to accuracy, and higher resolution images (e.g., 1024×768 pixels x 300 dpi) are better for OCR processing and give better accuracy than lower-resolution images. Color images result in better accuracy than grayscale or binary images because they contain more information that can be used by the OCR processing algorithm. Well-aligned document image acquisition or scanning may assist the OCR engine in interpreting text lines and blocks appropriately, and noise reduction strategies can prevent character misrecognition, a common problem in traditional or deteriorated documents [43].

Google Cloud Vision API exhibits better performance than conventional OCR engines such as Tesseract, particularly on complex scripts and noisy images. Post-processing techniques can be wrapped around the API for better results, such as applying domain-specific dictionaries or keyword lists for correction of error-prone symbols and recognition of specialized terminology. For documents with sparse or complex textual issues, fusion results of Google Cloud Vision API with other OCR engines utilizing string fusion methodologies may lead to extra improvements [43, 52].

The cloud-based API is self-scaling, running on optimized infrastructure across multiple data centers, thus offering the same level of performance and reliability even for individual and desktop users as it does for enterprise customers. Integration with other Google Cloud products supports end-to-end document processing workflows, but businesses should weigh the data privacy needs, reliance on internet connectivity, and ongoing operational costs when considering the service for their use cases.

3.2 ABBYY: Overview and History

Headquartered in Moscow, Russia, ABBYY is the global leader in OCR technology and NLP (natural language processing) software. It was established in 1989 by David Yang—a designer of the first Soviet windowing GUI who was pursuing education in the United States at the time—and was initially known as BIT Software (citation needed) and was selling a business version of a linguistic program Integrirovannyi paket programm Dianija. In 1989, the company introduced TextGrabber, the first international language support ABBYY soft-

ware product and FineReader—an OCR software for printed text [53]. Yang's vision was for students to go from one language to the other in a single click, from digital dictionaries to OCR and NLP technology [53]. In 2009, ABBYY's products were being used by over 30 million people throughout the world, demonstrating its wide application to industries including health, finance, and education [53].

The company's early years were marred by challenges presented by the post-Soviet economy, such as rampant software piracy (e.g., illegal copies of Lingvo greatly outnumbered legitimate sales) [53]. Nevertheless, ABBYY leveraged on its viral draw to drive brand recognition amongst companies that valued licensed software. BIT Software was renamed ABBYY in 1997, a word made up of proto-language "a" tag and "bit" tag, meaning "attentive eye", corresponding to the company's focus on accurate text recognition [53]. International expansion ABBYY opened its regional offices and partners across the Europe, Americas, Asia, and global sales became crucial for the company during Russia's financial crisis of 1998 [53]. Strategic investment, such as the support from Mint Capital in 2001, and redirection to a matrix-style organizational structure, allowed ABBYY to expand operations and innovative work in the areas of both consumer (B2C) and business-to-business (B2B) [53].

The AI company ABBYY has developed its technologies following the trends in AI, especially in OCR and NLP. Its Compreno platform, a semantic NLP system, is capable of contextual text comprehension and capable to provide intelligent search and high-quality translation [53]. 2009 saw ABBYY face increased competition from free solutions like Google Translate and, accordingly, a forced change in the business model, with freemium models explored [53]. Today, ABBYY still sets the standard for content capture and document automation, and is optimizing business operations for its customers globally [53, 54].

3.2.1 ABBYY's Key Products

Main Products of ABBYY ABBYY offers a complete range of software for OCR, NLP and data capture, as well as mobile and desktop products sweeping across the needs of individuals and organizations, regardless of their size. This review is summarised in the next section according to the references, highlighting the most relevant products.

3.2.2 Lingvo

Lingvo Since 1990 when ABBYY's first product Lingvo was introduced (see [53]), a first electronic dictionary featuring translations from Russian into English [53]), that company has made a network-corpora work out of machine-read and translation devices. It jumped the constraints of hard-copy dictionaries and was soon widespread despite extensive piracy – an estimated 50,000 illegal copies in 1991 alone [53]. Such success of Lingvo has greatly contributed to the promotion of the ABBYY brand, and it was included in Russian textbooks for English and German languages in 2007, thus affecting the educational area of application. Product development continued, and the software eventually learned to read several languages, making it today an important part of ABBYY's NLP portfolio.

3.2.3 FineReader: OCR Solution

OCR as a Solution ABBYY FineReader was released in 1993 and is one of the world's most powerful OCR software for converting scanned documents, images, and PDFs into editable and searchable text [53, 55]. It is available in a number of different versions, including FineReader PDF and Server, and it is known for its accurate detection and flexibility [54].

In the FineReader pipeline (cf. [54]) there are several stages such as pre-processing (adaptive binarisation, noise filtering), document analysis, recognition and export. Camera OCR enhances the processing of digital photographs [54]. The recognition algorithms are patented, and no detail is mentioned in the supported materials as to the use of CNN or RNN [55, 54]. New trends in OCR are the potential application of CNN for feature extraction and then RNN with recurrent temporal classification [56].

In [55], the accuracy of FineReader was 97.98% for PDFs, 96.98% for vehicle registration plates, 89.92% for receipts, but only 59.14% for handwritten and 37.5% for skewed. The profile `DocumentArchiving_Speed`, for example, supports up to 165 pages/min on multi-core CPUs [54]. It is suitable for archiving and form processing, it is robust enough [55, 53].

3.2.4 FineReader Engine SDK

FineReader Engine SDK, frozen at the beginning of 2001, enables the development of applications on the basis of ABBYY's Optical Character Recognition (OCR) and it supports

Windows, Linux, as well as Android [53, 54]. It utilizes tools for barcode reading, text orientation estimation, document structure analysis and many others making possible text extraction in more than 190 supported languages using flexible licensing models for both applications and embedded systems [54].

The SDK standard profiles, e.g., `TextExtraction_Accuracy` achieves high accuracy at which [55] reports 97.98% for PDF files, 89.92% for receipts, and since it decreases with the ratio of handwritten text (59.14%) and distorted images (37.5%) [55]. Parallel processing has a speed of up to 294 pages per minute [54]. Presumably, these are based on cutting edge algorithms such as CNNs or RNNs, although the details are not released [56, 55].

3.2.5 Confidence Scores as a Proxy Measure for OCR Quality

Confidence Values as a Proxy for OCR Quality The reliability from ABBYY FineReader, which is an indication of the probability of the right recognition of words or characters and one of the crucial measures in the assessment of OCR quality for large collection digitization. Cuper et al. [8] studied 2,000 pages of Dutch historical newspapers (1631-1995) OCRed with ABBYY FineReader (versions 8.1, 9, 10, and 12) and connected the confidence values with the word error rate (independent of order, WER_{oi}) to create an alternative measure of quality.

Page-Level Analysis

Analysis at the Post Level Cuper et al. [8] determined substantial strong negative relationships (Spearman's rho: ranging -0.760 7 to -0.934) between the WER_{oi} calculated with ocrevalUAtion [57] and the mean reliability of the words, except for the non-optimised configuration of manufacturer B (rho: 0.367). Three different approaches for building a WER_{oi} proxy were applied to 2000 pages OCR'ed with ABBYY FineReader 12:

In the recent OCR quality research (e.g. Cuper et al. [8]) are also exploring confidence scores produced by ABBYY FineReader, as a possible approximation for the real OCR accuracy. This is particularly important when no ground truth is available for large collections and institutions need a robust heuristic to prioritize re-processing. To simulate OCR error, the authors observed the correlation between average word confidence scores on a page level, and the Word Error Rate (WER*oi*), a version of WER that pays no attention to word order. Their experiments evaluate three types of conversions from word-level confidence to proxy WER*oi*: naive inversion, linear regression, and polynomial regression.

1. Naive Transformation The naive method simply inverts the word confidence wc to approximate the error rate:

$$\text{proxy} = (1 - wc) \times 100$$

Here, wc is the average word confidence for a page (on a 0–1 scale), and the proxy estimates the percentage of incorrect words. While simple, this method assumes a linear and inverse relationship, which may not reflect real OCR behavior.

2. Linear Regression This method models the relationship between average word confidence and WER_{oi} using a first-degree polynomial:

$$\text{proxy} = a \cdot wc + b$$

where a and b are coefficients determined by fitting a regression model on a training set. This allows for data-driven calibration, improving accuracy over the naive approach.

3. Polynomial Regression To capture non-linear patterns in the data, a second-degree polynomial is used:

$$\text{proxy} = a \cdot wc^2 + b \cdot wc + c$$

with a, b, and c fitted from the training data. In the study, this model achieved the best fit, with a mean square error of 17.75 and $R^2 = 0.793$. The optimal formula found was:

$$\text{proxy} = 276.234 \cdot wc^2 - 550.55 \cdot wc + 275.748$$

This polynomial regression method proved most effective in classifying pages into quality categories such as ”Good”, ”Average”, and ”Low” with a total accuracy of 83%.

Word-Level Analysis and Proxy Evaluation

Whereas page-level confidence gives a rough indication of the OCR quality, word-based analysis gives a more subtle means of both evaluating and locating recognition errors. This is especially helpful when you are processing high visual noise, complex layout, or mixed fonts within document segments. In recent work, word-level confidence scores have been compared against reference annotations to investigate how well they function as correctness indicators [58].

The ocrevalUAtion tool [57] provides word-level metrics that are crucial Today in order to determine if a recognized word is the gt is correct, partially correct, or incorrect. When the

OCR output is aligned to a ground truth substring on a word-by-word level, the frequency and type of the errors (e.g., substitutions, deletions, insertions) can easily be found.

In addition, word-level confidence scores are available to mark recognition which are believed to be uncertain, allowing other algorithms such as manual correction or a filter to be applied. E.g., it may be to highlight offer words with some confidence below 80%, etc., thereby allowing improved QA without mandatory re-OCRing.

Conclusion This integration of page- and word-level confidence analysis soundly determines an overall method for estimating OCR quality (without reference to ground truth). Page-level scores provide scalable heuristics for large document collections, and word-level scores are good for localizing errors. Regression models, in particular polynomial transformations, can be used to closely approximate OCR error rates by mean confidence. Backed by tools such as ocrevalUAtion, confidence-based proxy metrics are potent instruments for triaging poor quality text, prioritizing reprocessing, and enhancing the reliability of results in the real-life digitization pipeline.

3.3 Deep Learning based OCR

Although conventional, handcrafted, and rule-based methods have usually been employed in early OCR systems, the advent of deep learning has revolutionized the detection and recognition of texts, where convolutional neural networks (CNNs), recurrent architectures, and transformer-based models are now pivotal in state-of-the-art OCR systems.

3.3.1 CNN based Text Detection

CNNs now have been widely used in OCR for document processing, especially as text detection in natural scenes images and in structured documents. In this survey, we focus on the recent written characters' localization and recognition methods based on CNNs, including the most performant architectures, character localization methods, data augmentation and training strategy, performance improvement with attention mechanism, feature pyramid network, and complexity in terms of computational cost. The conversation is framed in terms of AI-based document processing in a context informed by recent developments and references shared.

Best CNN Architectures for Text Detection

CNN network structures such as ResNet, VGG, EfficientNet are important for text detection since they are strong for feature extraction. **ResNet** [7]: ResNet uses residual connections to alleviate vanishing gradients and facilitates deep networks such as (e.g., ResNet-50, ResNet-101) which are very successful in text spotting in complex natural scenes. Due to its depth, images with fine details can be captured, which is applicable for acquisition of documents with a noisy background (e.g., legal or historical text) [8]. **VGG** (e.g., VGG-16) utilizes a shallow architecture to calculate the number of filters in each layer, with smaller 3x3 filters which balance execution time and solution quality. This model is useful when applied to structured documents with meaningful text and layout structuring. [12] However, this method may not work well for the highly varied text [13]. **EfficientNet** scales the model in depth, width, resolution, and it has improved convergence rate and accuracy with fewer parameters, suitable for a variety of document types (multi-lingual, handwritten) [15, 16].

RPN vs. Single-Shot Detections

Comparison of localization methods Text can be localized in documents using RPNs or single-shot detectors. **RPNs**, the region proposal networks are two-staged as the mentioned Faster R-CNN and CTPN: region proposals generation and classification; regression refinement. This technique has very high precision for complex layouts of document pages like piping and instrumentation diagrams or legal documents with heavy text density [17, 5]. RPNs are, however, computationally demanding and they may not be suitable for real-time use [6]. **Single-shot detectors:** Single-shot detectors (such as the EAST and YOLOv8) predict text regions in one forward pass, providing faster inference and are more appropriate for real-time document processing [18, 15]. They are essentially less accurate in very dense or cluttered text cases and computationally more efficient [4].

RPNs yield higher precision (e.g., F1-scores of 0.90+ ICDAR) on complicated documents, and single-shot methods prioritize speed (e.g., 10-30 ms vs. ICDAR) [18, 24]. The latter have gained a lot of attention for AI-based document processing and specifically, single-shot detectors (EAST, CRAFT-YOLOv8) as they are a reasonable compromise in terms of speed and accuracy especially for online-video real-time extraction, or text recognition of scanned forms or IDs [48, 16].

3.3.2 CRNN in Text Recognition

CRNN is a fundamental architecture in text recognition, which combines CNNs' strength in spatial feature extraction and the sequence modeling ability of RNNs. The performance of CRNN systems is greatly influenced by the options of CNN backbone, RNN variant and addition of more sophisticated mechanisms like attention and connectionist temporal classification variants.

The CNN backbone for feature extraction is one of the key factors that affect the recognition performance of the CRNN systems. ResNet-18 has become a very popular backbone to be used because of these residual connections which would help to retain the extracted feature information and thereby reduce the loss. According to literature, replacing standard VGG-16 convolutional architectures with ResNet-18 within CRNNs improves recognition and feature retention, especially when dealing with challenging framed scenarios, such as those with complex backgrounds or distorted text [59, 60]. VGG-16 is also a strong candidate, especially when pre-trained on large datasets with good performance on multilingual and cursive text recognition tasks [59, 60].

For resource-constrained applications, MobileNetV3 provides the best trade-off between accuracy and efficiency, maintaining competitive performances with lower model sizes and computational complexity [61]. Novel hybrid approaches, such as MobileViT [62], leveraging lightweight Transformer blocks inside CNN backbones, build on top of this progress, improving the context modeling and feature extraction even further, without compromising the parameter efficiency. The choice of the specific type of RNN — i.e., whether LSTM, GRU or a bidirectional structure is employed — significantly impacts the recognition accuracy and training speed. Both forward and backward architectures consistently achieve greater accuracy as well as context by taking into account future and past sequence information. BiLSTM and BiGRU have achieved an accuracy above 90% in the handwritten text recognition where the accuracy for BiGRU is 90.32% versus BiLSTM 90.04% [63, 64].

Although from the accuracy point of view the LSTM-based CRNNs are slightly better with F1-scores of 98.37% (compared to 94.84% for the GRU-based models), the computational trade-offs favor the GRU models in the context of efficiency-critical applications. GRU has a lesser number of parameters and faster training times and is efficient for the IBT [65, 66]. Mixed BiGRU and GRU encoder-decoders systems also yielded favorable results, leading up to 6-10% of relative improvement in character and word recognition rates along

with 18 seconds of time reduction per training epoch in comparison to BiLSTM-LSTM networks [65].

Apart from the classical CTC loss functions, in recent years, attention mechanisms and hybrid training strategies have shown to be an obvious advantage to the overall results of the CRNN. By training the model with joint CTC and attention, the strengths of both approaches are combined, leading to the most competitive word error rates of 0.014 for word recognition for domain-specific tasks such as license plate recognition [67]. CRNNs with multi-head attention mechanisms (MAH-CRNN+CTC) can perform targeted feature attention and more effective data segment weighting and achieve 91.2% in recognizing characters in technical documents [68].

These CRNN networks provide strong performances on different text lengths, font diversities and image variants. To explore higher computational capacity (as illustrated in [69]), feature extraction in multi-scales and the attention mechanism, the enhanced CRNNs, give birth to fast and accurate algorithms for short- and long- text sequences, however, is only robust to text with high density [69, 70, 60]. Multi-font training techniques obtain low character and word error rates proving good generalization to never seen before font styles [69, 59]. The issues of image quality are well-treated by incorporation with super-resolution approaches and image enhancing blocks. CRNNs with local entropy feature extraction and super-resolution techniques provide up to 10.6% accuracy improvement on challenging datasets of low-resolution and noisy images [71, 72, 73, 74].

The success of a CRNN is based on the smart combination of real and synthetic training data. Combining two sources of data in the form of hybrid datasets decreases the need for real data by 80% and increases the performance of the model by 16% [75]. Synthetic data synthesis using the GANs, VAEs, and 3D modeling technologies give the alternative data generation mechanism, which gives the variety and more well-annotated datasets leads to better model performance even in the less data availability domains [76, 77].

3.3.3 Vision Transformers (ViTs) for OCR

Recently, attention-based Vision Transformers (ViTs) have been proposed as an alternative to traditional convolutional networks and have achieved significant improvements in OCR. Trained on image classification tasks over fixed-sized patches and single-label assignments, ViTs have been considerably modified to handle the specific issues and constraints of

text recognition, such as variable text layout, multi-scale detection, and sequence prediction.

On the other hand, decoupling belongs-to-graph can lead its transition from image classification to OCR, and the standard ViT architecture has to be redesigned fundamentally. As originally designed, ViTs treat images as sequences of uniform fixed-sized patches, and then directly output a single class label based on global feature accumulation. For OCR work, these models have to be combined with text localization and sequence prediction [78]. Additional improvements consist of the utilization of transformer decoders allowing direct prediction of text region bounding boxes without utilizing extra post-processing steps, which allows for end-to-end text detection frameworks. This architectural feature enables combined detection and classification similar to the differentiation between handwritten and typeset text in the same model [78]. The modified ViT architectures also focus on local and global feature learning that correspond to the 2-fold processing (fine-grained character and more abstract context) needs in OCR. Although the ViT encoder remains strong at capturing global context through self-attentions, the architectural changes make it able to keep the small details that are required for accurate character recognition and to use them in the lower stage [78].

To achieve good performance of ViT for text recognition, achieving the best patch embedding strategies is more complicated than applying straight uniform grid approaches. Text-aware patch alignment (PTA) is a major step forward and results in patch embeddings that are aligned with real text regions instead of bribing the approximate grid divisions. Such a method can decrease computational burden by discarding superfluous patches, yet keep or improve the recognition performance [79]. Heterogeneous/weighted mixed-embeddings: Advanced methods of embeddings (heterogeneous, weighted) based on mixed type embeddings, such as spatial gated embeddings, Fourier token mixing, and MLP mixture embeddings. The weighted sum of these different types of embeddings, with special emphasis on the spatial gated embeddings, has been shown to achieve better text recognition [80]. Learning position encodings Geometric and fractal position encodings confine spatial localities in patches such that the closeness of patches would be preserved, thereby preventing much less loss of information than standard sliding-window mapping. This preservation of spatial relations between samples results in higher accuracy and faster convergence during training [81]. Input pre-processing methods also contribute to the performance of ViT by involving CNN-based feature extraction to replace traditional patch embedding and with a few initial feature processing using convolutional layers. This hybrid local-detailed concatenated

model can enrich the local feature representation and is particularly useful for recognizing handwritten or small text [82]. Span masking regularization, a method that masks neighboring features across channels in feature maps, is one of the effective regularizers that improve generalization, particularly useful with small or noisy datasets [82].

The CNN-ViTs blends CNN and ViT architectures, which turns them to be synergistic, because they benefit from the complementary properties of both methods. CNNs are good at capturing fine-grained local features, such as edges, textures, character-level patterns, and so on, while ViTs are better for global context modeling based on self-attention and considering the distant relationship of the image blocks together for the whole image [83, 84, 85]. Most of these hybrid architectures process CNN layers at the input level for local features and apply transformers for global context. In this chained pattern, both character-level and document-level information can be naturally incorporated [83, 84, 85].

Multi-scale/hierarchical processing stages in cascade architectures of hybrids guarantee the preservation and good integration of information over various spatial resolutions. The mixed convolutional feed-forward layers promote feature learning at both the local and multi-scale levels, thereby achieving more accurate predictions than pure CNN or ViT methods [83, 84]. The efficiency advantage of hybrid architectures makes them particularly appropriate for real-time and mobile OCR systems, whereas their local and global processing mixture brings better robustness against noise, distortions, and various text styles [83, 84].

ViTs have unique data and training needs that are distinct from conventional CNN methods. This lack of built-in local inductive biases makes ViTs more data-hungry, generally demanding large and diverse datasets in order to reach state-of-the-art performance. This attribute presents special challenges in OCR for diverse text styles as well as complex layouts. ViTs typically lag behind CNNs on smaller datasets unless additional techniques are applied. Area-specific information-enhancing modules, like Local Information Enhancer assist ViTs in understanding fine-grained details and achieving better generalization in a small data setting [86]. Self-supervised and auxiliary training tasks have been shown to be an effective approach to dealing with data scarcity. Simultaneously optimizing for primary OCR tasks and self-supervised objectives (e.g. masked image modeling or translation perceptibility) can further improve performance, especially as the labeled data is relatively limited [87, 88, 89].

Data augmentation strategies should be well thought out for ViT training as ill-posed augmentation may induce variance shift in positional embeddings. To this end, methods such

as Mixup, Cutmix, random erasing [90], etc., should be tailored to the ViT architectures not to cause embedding-related issues in enhancing generalization [91].

Domain generalization approaches transfer the token-level feature stylization (TFS-ViT) which combines the normalization statistics from two different domains, improving ViT generalization capabilities on unseen data and domains, which is important for practical OCR scenarios across various document types [92]. Specialized components could be added to ViTs to deal with the problem of various document layouts, multiple text orientations, and multi-scale text detection. Grid-based tokenization mechanisms, as adopted in models such as Vision Grid Transformer (VGT), adopt two-stream architectures with the token-level and segment-level to learn visual and semantic representations, thus facilitating accurate prediction of complex multi-region document page layouts [93, 85]. Two-dimensional relative attention mechanisms enable ViTs to pay attention to the spatial relationships between tokens, hence are promising for multi-page and arbitrarily organized documents. Such spatial information allows the model to know document structure and not only sequential relations [94].

By these, they can address arbitrary text orientations by taking advantage of their inherent capability to model long-range dependencies by means of self-attention mechanisms, which allows achieving resilient text recognition irrespective of orientation. Some ViT-based models have such abilities to predict the bounding boxes directly, and directly localize and classify the text in arbitrary orientations, without additional post-processing [95, 78]. The issue of multi-scale text detection is tackled by using adaptive patch and n-gram embedding schemes that adaptively modify patch sizes to cope with text scale variations in images. More advanced models such as ANT-STR leverage these adaptive embeddings to learn effective text snippets on small and long text segments [96].

Dynamic scale fusion modules fuse multi-scale feature maps for better detection for texts of varying sizes within the same document or scene. Diagonal attention mechanisms, such as DiagSWin attention, allow tokens to attend not only over nearby regions but also at various levels of resolution, to enhance multi-scale context capture and detection quality [95, 97].

3.4 Techniques for Accelerating Document Processing

Use of the most advanced OCR systems in real applications requires not only high accuracy but also the possibility to process in real-time large volumes of documents. With organizations drowning in millions if not billions of documents to digitize, and the need for real-time text extraction for use cases as diverse as indexing documents on the fly to extracting text from live video, the computational requirements of advanced deep learning models are a major bottleneck. Recent OCR systems need to manage a trade-off between the complexity necessary to recognize text with high quality in different types of documents, together with the speed and resource limits in a production environment.

3.4.1 GPU Acceleration

The use of modern OCR systems on GPU computation models has completely transformed the landscape of document processing and is already yielding huge performance gains that make real-time, high-throughput text recognition practical for production use-cases. CPU-based processing to GPU acceleration is one of the key improvements in OCR system efficiency, which gives such a level of performance that unthinkable applications and processing sizes become feasible.

Recent NVIDIA GPU architectures offer huge acceleration potential of OCR workloads with scaling performance advancement compared with the traditional CPU-based implementation. For example, work that used NVIDIA P100 GPUs demonstrates that parallelizing the preprocessing operations of OCR operations that lead to an average image processing latency of 48.32 ms on CPU can be sped up by a considerable factor (0.825 ms) on GPU, achieving an impressive 58.6x speedup over the CPU execution across processing on the CPU [98].

Although direct comparison studies of specific architectures such as V100, A100, and RTX are scarce in the literature, the massive acceleration presented by the old P100 architecture implies that newer GPU generations hold even more potential. The V100 and A100 architectures have a number of improvements such as Tensor Cores, higher memory bandwidth, and CUDA core counts which are particularly useful for the matrix operations and parallel calculations in the recent deep learning-based OCR systems [98].

Though the RTX series may be more targeted towards consumer hardware, its accessible supercomputing capability, with Tensor Core support and improved FP16/INT8 operations,

keeps advanced OCR acceleration possible in wider usage situations. These architectural enhancements allow recognizing with GPU-based OCR engines large-scale workloads and multiple data streams concurrently and are ideal for enterprise document processing applications [98].

To fully utilize the GPU in OCR, advanced memory management strategies which overcome the capacity constraint of GPU memory and bring the computation close to peak throughput are needed. Recently presented memory saving techniques based on on-the-fly tensor swapping and recomputation became critical elements when optimizing memory utilization for GPUs.

The intelligent tensor swapping systems, such as pommDNN and the HOME, employ the optimization algorithms (e.g., genetic algorithm and particle swarm optimization) in the determination of tensor placement and swapping schedules between CPU and GPU memory. These techniques make possible selective recomputation of intermediate results to minimize GPU memory overhead, and allow us to use larger batch sizes, leading to higher overall throughput. On the other hand, throughput is improved by up to 57% and the maximum batch size by up to 2.8x on representative LSTMs vs the naive memory management approaches [99, 100].

Fine-grained memory optimization strategies, as can be found, for example, in DELTA, which have both bidirectional prefetching and fine-grained memory management, can save up to 40-72% memory usage and support batch sizes 6× as large as naive baseline methods. These memory reductions have been achieved without sacrificing throughput, thereby enabling the processing of larger documents, or more concurrent OCR tasks, within the same hardware budget.

In order to fully utilize the performance of GPUs on OCR workloads, it's needed to have dynamic batching processing strategies according to the characteristics of the real-time systems and the workload. Dynamic batch sizing strategies like we implemented in BatOpt systems, change the batch sizes to meet the current system state and arriving task rates, balancing the service latency requirement and GPU memory usage limited.

Our adaptable batching strategies can offer 31x the reduction in latency achieved using a single-input approach with 4.3x the performance of a fixed-batch approach while consuming only a fraction of the memory required by greedy batching strategies. Dynamic optimization of batch size for turning requests is an effective way to utilize resources and facing the varying

OCR workloads more responsively [101].

Mixed-mode resource management systems (mixed-mode RMs) for hybrid systems have the flexibility to allocate resources for the batch and streaming systems, and the ability to share the GPU resources among workloads with an even finer granularity. They are especially useful in mixed workload scenarios where the OCR workload needs to coexist with others, achieving the best resource usage for applications with disparate requirements [102].

Novel resource sharing techniques employing shared memory between processing elements and dynamic distribution of GPU resources using workload monitoring achieve up to 134% higher throughput, as well as lowering overhead due to data transfers. These performance-oriented optimizations are especially crucial in OCR-centric applications with several processing stages and frequent data communication across the different computational blocks [103]. Predictive batch scheduling schemes that predict both memory demands, schedule the requests in both perceptual and predictive batches and alleviate the interference between CPU and GPU operations. This APO achieves better overall performance and resource utilization in heterogeneous computing environments because the OCR processing is necessary to manage complex memory hierarchies and balance among multiple processing units [104].

Mixed-precision methods are an important technique for optimizing OCR systems, and using lower (FP16) and standard (FP32) precision arithmetic can lead to a significant performance gain with little loss of accuracy. These methods can further offer a speedup in both training and inference up to 1.9x on GPUs with respect to typical FP32 computations, gaining up to 3.74x when leveraging distributed or parallel training approaches [105, 106, 107].

The memory benefits are similarly crucial and the GPU memory footprint is reduced by 29-50%, allowing for larger number of samples in a batch and more efficient utilization of hardware resources. We achieve such memory savings, which can be immediately converted to higher throughput capacity for OCR to process images per second, without the need to upgrade the hardware [108, 106, 109, 110]. The preservation of accuracy in mixed-precision OCR systems is achieved through a fine-tuned design approach that exploits the resilience to small numerical perturbations brought about by lower precision arithmetic in deep neural networks. Advanced from layer or even operator-wise mixed-precision techniques further reduce the loss of possible accuracy, in the case that a lower precision can be selectively chosen for those operations, which does not degrade convergence or final recognition result [106, 109, 110].

The synergy of these optimization strategies –from high-performance memory allocation and automatic batching, to mixed-precision computation– empowers modern GPU infrastructures to meet the performance needs of real-world, large-scale OCR deployment. These optimizations are important across a wide range of uses, from live document scanning to batch processing of digital archives, ensuring that advanced OCR functionality is available without regard to the computational environment or usage characteristics of the application.

3.4.2 Parallel Processing

Parallel processing has become one of the fundamental means to improve OCR system efficiency in the context of substantial document processing tasks. The success of parallel processing in OCR tasks is determined by a wise selection of frameworks, parallelization mechanisms, and optimizations, and it can dramatically affect throughput and system latency.

For distributed OCR processing, some parallel processing systems have proved effective. Apache Spark and Flink are examples of popular systems for parallel distributed data processing. These systems provide powerful support for task scheduling and resource management. These schedules can also be improved by adopting advanced scheduling algorithms like network load variation perception and dual-phase pipeline task schedulers, which improved resource utilization and reduced processing time in OCR pipelines significantly [111].

Combining Apache Hadoop with TensorFlow for this merging of distributed storage with deep learning has shown good results, especially with OCR processing in a scalable, parallel manner. This joint has shown significant enhancement of performance, increasing OCR effectiveness by 20% and efficiency by 30% compared to a traditional model [112]. These results demonstrate the potential of distributed computing architectures for satisfying the resource requirements of contemporary OCR applications.

Another important progress in DML for OCR applications is FSP frameworks analogous to Flegel. These frameworks aim for distributed speedup while simultaneously minimizing the synchronization overhead by wasting the minimal amount of workers [113]. Additionally, for tasks such as OCR, which are widely used for large-scale text classification, symmetric ADMM (Alternating Direction Method of Multipliers) makes parallelization of the algorithm across multi-node clusters computationally less intensive, improving the scalability and convergence [114].

Various parallelization schemes—data, model, and pipeline parallelism—impact OCR

throughput and latency differently. Data-parallelism that spreads different images or batches of data across a number of processing units enables simultaneous processing of a large number of images so that overall throughput is greatly increased. This is even more pronounced with latency reduction: for example, GPU supports streaming OCR pipeline where both average delay and residual delay dropped from 48.32 and 35.3 respectively on a CPU down to 0.825 and 0.75 ms on GPU, thereby obtaining the impressive 58.6x speedup over serial execution [98].

The OCR pipeline parallelism splits the OCR processing into consecutive stages, which are implemented by different processing elements (PEs). This method even allows for data to flow without stopping, thus achieving optimal hardware usage for big workloads to further improve throughput. Pipeline parallelism increases throughput by overlapping the execution of different stages, eliminating idle time and reducing the overall processing latency for each image stream [98].

Efficient use of multi-core CPUs for OCR requires advanced load balancing and task scheduling algorithms. In this context, AI-based scheduling methods have been especially promising. Scheduling methods such as GG-DRL adopt Deep Reinforcement Learning (DRL) to model task scheduling as a Markov Decision Process (MDP) and use neural networks to learn the optimal task-to-node assignment. Such methods have the advantages of adapting to the dynamic task dependencies and heterogeneous processors more naturally than the static mapping-based ones [115].

Distributional reinforcement learning methods further have the advantages of modeling the entire distribution rather than only the first moment. Through this approach, we can achieve a more robust scheduling mechanism in the face of dynamic and stochastic workloads, that gives better cluster load balancing, task success rate, and average completion time, and also offers good scalability to large, real-world traces [116]. Performance knowledge scheduling in real time with SAC-based algorithms is able to dynamically identify and recover from the variance of server load in order to minimize the variance of task load and the average response time, especially at high or random loads [117].

Metaheuristic algorithms also play an important role in load balancing optimization. The OIWHO algorithm utilizes learning and chaos perturbation strategy to achieve a balance between exploration and exploitation for optimization search. It is able to have shorter time to complete tasks than others EAs, such as PSO, and genetic algorithms, especially in the large-

scale scheduling problems [118]. Similarly, the Balanced Prediction Priority Task Scheduling (BPPTS) algorithm considers both the computation cost and the priority of a task in order to provide a better solution in terms of load balancing and scheduling efficiency for compute-intensive OCR tasks. It outperforms traditional list scheduling algorithms in terms of completion time and speedup [119].

Asynchronous processing and queue management systems are very important to increase OCR system responsiveness when dealing with big numbers of documents or real-time requests. Independent operation of tasks by asynchronous approaches helps to increase throughput and reduce the user waiting times and is particularly effective in real-time and bulk document processing applications [120].

The components of the OCR (pre-process, recognition, post-process) are able to work independently and in parallel, and system response time performance can get to be reduced by a short time [120]. Such asynchronous approaches may be nicely augmented with a more sophisticated queue management process that optimally handles incoming OCR requests, redistributes work, and gives priority to high-priority or time-sensitive work to be performed in a timely manner with low latency.

Advanced queue management schemes like the TrioPen framework employ multi-level penalties to encourage flows with low latencies and to discourage flows with high latencies. This successfully minimizes end-to-end delay, jitter, packet-loss, and enhances throughput to respond to the sensitive task – such factors are essential to achieve real-time OCR responsive [121].

These parallel processing paradigms are integrated to form a hybrid framework for OCR acceleration which considers both computational efficiency and system responsiveness. The co-operation of distributed platforms, clever scheduling algorithms, and various asynchronous processing technologies allows new-generation OCR systems to tackle the growing volumes of work with high accuracy and low latency.

3.4.3 Edge Computing Solutions

With the advent of edge computing, OCR can run as a transformative service on the edge, which brings real-time text recognition capabilities on low-capability devices and decreases the delay time and dependency on a stable network connection with the cloud. To develop OCR on the edge, one needs to pay attention to model compression, platform-specific opti-

mization, distributed training, and robust offline process.

To deploy OCR models in an edge device, we need to have advanced model compression methods to greatly shrink its size and computations while maintaining high accuracy. The work in [122] has shown that knowledge distillation is the most effective single technique with up to 11.4x reduction in model size and 78.7% speedup in latency with only a moderate drop in accuracy, which is well-suited for preserving OCR performance on resource-limited edge devices.

Another successful approach is pruning and quantization for deployment on edge OCR. Bi-medial based frameworks that integrate both methods can decrease the model memory by up to 94.9% with no more than 3% loss in accuracy. The combination of these approaches is particularly useful to fit OCR models under very limited hardware constraints without sacrificing the quality of the recognition [123, 124].

Reinforcement learning-guided automatic pruning methods, e.g., Multi-Constrained Model Compression (MCMC) can achieve even more aggressive up to 80% FLOP reduction with memory saving. Surprisingly, these methods can sometimes even obtain improvements over baselines, indicating that there is room for the compression to make the model better instead of making it worse [125].

Mixed-precision quantization methods have more beneficial properties for edge OCR deployment. Ultra-low mixed-precision quantization methods and architectures, such as 1-bit-backbone networks and use of 4-bit classification heads provide up to 16× model compression with high detection accuracy makes them appealing for real-time OCR applications on edge devices [126].

Various edge computing platforms have different strengths and optimizations to deploy OCR. The NVIDIA Jetson family also runs a variety of model formats like TensorFlow Lite (TFLite), ONNX, and TensorRT and is especially well suited for real-time, GPU-accelerated OCR. Google Coral platforms are designed to accelerate TFLite models for the Edge TPU, offering outstanding low-power, fast inference for Edge TPU-optimized models. The Intel Neural Compute Stick (NCS) can be deployed as a standalone USB plug-and-play device and supports deployment in both TFLite and OpenVINO formats which is suitable for low-powered devices [127].

Choice of model architecture is really important for performance between these platforms. CNN-RNN models, such as PaddleOCR and master-based models (e.g., UltOCR), suffer little

decrease in accuracy when converted to TFLite and quantized, and thus are excellent candidates for deploying to a wide array of Jetson, Coral, and Intel NCS devices. On the other hand, Transformer-based models such as TrOCR suffer significant accuracy degradation, up to 6.81% for sentence recognition post-conversion, thus requiring further fine-tuning to be effective at the edge [127].

Federated learning methods make it possible to train OCR models on edge devices in a distributed manner by enabling collaborative model optimization without sharing raw data. This model maintains privacy and regulatory compliance but also makes use of the distributed computational power on multiple edge devices [128].

The non-IID problem due to the heterogeneous data distribution of the edge devices is tackled by the federated learning framework with the following techniques: dynamic data queuing and entropy-based participant selection quality-aware aggregation. These methods offer an effective way to reduce loss of accuracy inherent in a heterogeneous distribution of data and increase model convergence [129, 130, 131].

Hierarchical frameworks and edge aggregators that allow for local aggregations before sending the results back up to central servers are used to communicate efficiently in federated OCR training. Such an approach is effective in reducing bandwidth requirements and speeding up the training processs [132, 133, 134, 68]. Adaptive participation policies (e.g., device selection dynamically, based on reinforcement learning for optimization) can further improve the efficiency by guiding which devices participated and at what frequency, and they also can address issues regarding device heterogeneity and computational straggler [130, 133, 131, 68].

The reward-based aggregation combined with data-centric client selection models also ensures that edge devices with either better quality or diverse datasets contribute more to the model update and consequently we get enhanced OCR performance overall [130, 135].

Efficient offline OCR on edge needs powerful local inference systems that can process whole OCR pipelines (including text image preprocess, feature extraction, and character recognition) without relying on external servers. This guarantees real-time performance and robust operation in case of network discontinuities [126, 136].

It is important to develop efficient algorithms to support the offline processing. With lightweight scale- and rotation-invariant feature extraction techniques, accurate character recognition can be achieved with minimized computation overhead, making offline process-

ing possible even on applications with extremely constrained resources [136]. Advanced task scheduling schemes can maximize the tradeoff between local processing and potential offloading, which can significantly enhance efficiency and responsiveness in offline cases [126].

Local storage solutions need to trade off identification accuracy and storage space. Feature glossaries, which store compact and quantized local feature vectors instead of raw images or full features, promise to achieve both savings in storage and recognition accuracy. Such evidence glossaries allow for fast matching and retrieving of evidence in the offline-based recognition task [136]. Therefore, anticipating the possible data or model components required by the user in advance cannot be overemphasized to reduce latency and maintain seamless operation in a network-less environment [126].

Selective data preservation methods concentrate on preservation of key features or evidence vectors, producing feasible savings on locally stored data but also maintaining strong OCR capability (see for instance [136]). This approach guarantees that edge OCR systems can work efficiently with limited hardware capabilities and lossy connection.

3.5 Metrics for Performance Evaluation

OCR systems should be evaluated using a set of measures that can optimize their performance in various levels of grain size and different applications. Good OCR evaluation involves character-level precision, word-level accuracy, comprehension at the document level, and reliability with confidence scores. This holistic approach helps evaluate the practical application and deployment readiness of OCR systems.

3.5.1 Character-Level, Word-Level, and Document-Level Accuracy Measures

OCR assessment needs metrics designed for varieties of text granularity which tell different stories about system behavior and usefulness.

The Character Error Rate (CER) is a simple and yet powerful metric to evaluate fine-grained recognition, capturing the errors (insertion, deletion, and substitutions) in characters needed to change the OCR outputs into the ground-truth transcriptions. CER is especially useful for assessing OCR performance on languages with complex scripts, character-level

accuracy being essential for downstream applications [137, 138]. Character accuracy, which is expressed as the percentage of the correctly recognized characters, complementarily reflects another aspect of the CER and is an intuitive measure of the recognition performance [138, 139].

The Word Error Rate (WER) is the main metric used for word-level evaluation and measures how many word-level insertions, deletions, and substitutions are needed to match the OCR recognized output with the reference text. This is also because WER has a direct impact on practical text processing tasks and it is widely used not only in document text OCR but scene text OCR [20, 137, 139, 140]. The Bag of Words Word Error Rate (bWER) [137] provides a complementary view in the sense that we ignore word order, but instead compare only the set of words that have been correctly recognized, adding a degree of analysis of the intrinsic word recognition capability irrespective of sequence correctness. Word Recognition Rate, the accuracy of recognizing words, is an intuitive performance measure often used for practical OCR applications [139].

Document-level evaluation also leverages higher-level text understanding and layout correctness beyond character and word recognition. Page-level (or document-level:) WER Equation captures the effect of all recognition errors on entire documents, giving an insight into the overall system's performance on whole texts [137, 140]. Reading-order accuracy is one of the most important evaluation metrics for end-to-end document transcription because it reflects the quality of preserving the right position ordering of text lines and text blocks in images. Reading order preservation is formalized with advanced metrics like NSFD [141, 137].

Secondarily, two-fold evaluation approaches include some measure of transcript accuracy (WER, bWER) and reading order evaluation to provide a complete evaluation of document-level performance [137]. It ensures that OCR systems are evaluated not only for their recognition quality but also for their ability to preserve document layout and coherence, a necessary property for interactive document digitization tasks.

3.5.2 Benchmark Datasets and Benchmarks

Fair, reproducible, and complete performance measurement across different scenarios, languages, and application fields can hardly be achieved without standardized OCR data and tools. OCRBench is a crowdsourced evaluation for OCR in multimodal models, which is a large-scale benchmark that contains 29 datasets. This benchmark includes the listed tasks of

text recognition, scene text visual question answering (VQA), document VQA, key information extraction and handwritten mathematical expression recognition. OCRBench enables multi-lingual evaluation and handwritten text evaluation, releasing it publicly as a research tool [142].

OCRBench v2 augments the original framework with 4x additional evaluation tasks over more than 31 different types of scenarios that include street scenes, receipts, mathematical formulas, and technical diagrams. This extended benchmark offers 10,000 human-verified question-answering samples with a high percentage of challenging ones, which comprehensively tests the textual localization & layout perception and logical reasoning skills [143]. Another comprehensive evaluation benchmark is CC-OCR comprising 39 subsets and 7,058 fully annotated images spanning multi-scene and multilingual text reading, document parsing, and key information extraction. This benchmark has a large proportion of real-world application data, which makes the evaluation of OCR systems practical [144].

KITAB-Bench is tailored for Arabic OCR evaluation and contains 8,809 samples in 9 domains and 36 sub-domains. This task meets the different challenges of Arabic script extraction and document layout analysis that include handwritten text, tables, and charts, which are typical for Arabic documents [145].

HHD-Ethiopic is a historical handwritten Ethiopian script dataset which is designed specifically for Ethiopic script recognition experiments and comprises 80,000 annotated text-line images sampled from documents spanning the time period from the 18th to the 20th century. This dataset facilitates in-distribution in addition to out-of-distribution scenarios, for evaluation of OCR adaptation from one historical document varied to another one [146].

TCGA-Reports is a machine-readable dataset containing 9,523 reports from the pathology of cancer patients, which allows domain-specific benchmarking of OCR and text-based AI models on health topics for applications where accuracy and trust are fundamental for clinical outcomes [147].

REBU-Syn is a large-scale dataset containing 6M real and 18M synthetic samples, provided to evaluate scene text recognition methods and scaling law studies such as the performance of OCR under varying data regimes [148].

Dynamic Video OCR Benchmark responds to the new demand for OCR evaluation in the temporal setting, releasing an open-source dataset consisting of 1,477 annotated frames coming from a wide variety of video sources such as news broadcasts, YouTube, and code editors.

This dataset can help the research community evaluate OCR systems in mobile scenarios; it is of utmost importance to improving dynamic environments with reasonable temporal consistency and real-time performance [149].

Together with the standardized datasets and benchmarks, they contribute to the construction of a strong OCR evaluation baseline, which is conducive to fair evaluation across different languages, scripts, document types and practical use cases of the OCR works. The richness of benchmarks enables OCR systems to be evaluated in depth to match the deployment environment.

3.5.3 Confidence Score and Uncertainty Quantification

Confidence scoring and uncertainty quantification are important parts of OCR evaluation, allowing the estimation of the prediction quality and making informed decisions in the post-processing stage. Modern OCR engines generally output confidence scores for every recognized character or word, which describe the confidence of the OCR system in the correctness of its predictions. Such scores act as powerful cues for filtering low-confidence outputs which would need human inspection or one more step of processing on top [150, 58]. The utility of confidence scores as quality indicators relies on the fact that low confidence scores can be strongly correlated with high error rates, and when this correlation is strong, confidence scores can be practical surrogate measures of OCR quality [150, 58].

The usefulness of confidence scores is naturally tied to the OCR engine and its tuning parameters. The relationship between confidence and recognition is different for some engines for recognition performance to be calibrated over all operating points in order to get meaningful confidence measures to be used in ASR [58].

State-of-the-art uncertainty quantification techniques improve OCR error detection by incorporating confidence scores in complex models like BERT-based models. This integration greatly enhances the effectiveness of post-ocr-correction processes, because they can now perform a much more reliable detection of probable recognition errors [150].

Quality proxies that capture aspects of the relationship between confidence and error can give rise to automated text quality categorization systems. This can help in making an informed decision on the need for re-processing or targeted post-processing interventions, and thus to tune the right balance between the processing capacities and the output quality [58].

Engine-specific confidence scores and uncertainty quantification techniques offer influential but engine-dependent signals of confidence in OCR predictions. With adequate calibration and in the scope of comprehensive evaluation frameworks, these methods allow effectively detecting errors, evaluating quality automatically, and developing post-processing strategies for targeted applications with large impact on overall OCR system performance and robustness [150, 58].

The addition of a confidence score to conventional accuracy measurements gives a fuller understanding of OCR system performance, thus offering users the ability to assess the reliability of neighbors and the necessity of verifying or correcting the recognition outcome. Such a holistic approach to reliability assessment is especially relevant in sensitive applications due to the potential impact of OCR errors.

3.6 Challenges and limitations of the current systems

Although OCR technology is a remarkable industry, many problems and shortcomings have not been eliminated, which has an adverse impact on the accuracy, performance, and the application of OCR in various fields. It is important to understand these limitations to know under which circumstances we can paint, and what possible research and deployment issues need to be addressed.

3.6.1 Less-Resourced Languages and Non-Latin Script

OCR systems struggle with particular intensity when dealing with low-resourced languages and non-Latin scripts, posing obstacles for digital inclusion and global access to OCR. Low-Resource OCR: The primary problem in OCR for low-resource languages is the lack of annotated training data. The creation and processing of low-resource language-wide and high-quality annotated datasets necessary for training modern OCR systems is a challenge which makes the development and benchmarking of OCR models arduous [151, 152, 153, 154, 155, 156, 157]. Even though synthetic and augmented data can help solve these deficiencies, it is hard for synthetic data to represent the diversity and subtleties of real-world documents, especially on complex script and handwritten material [151, 155, 157].

Non-Latin alphabets have an intrinsic structural complexity that makes it much harder for OCR to process. In addition, these scripts like to utilize stacking, diacritics, ligatures,

non-uniform character widths, and writing systems that do not represent word boundaries explicitly, which largely increases the difficulty of both the operation of segmenting and the operation of recognition themselves [156, 158, 159]. Several scripts have visually confused display characters that lead to misrecognition, particularly in the presence of limited training data [154, 156, 159]. The problem is exacerbated due to variability in handwritten linguistic texts and aging of historical documents [151, 155, 157].

The OCR models typically trained for Latin scripts do not work well for non-Latin scripts because they possess different fundamental structures and do not have the transferability [151, 160, 153, 156]. BayesC -π models are being developed to handle these specific challenges, however, these are domain-specific and still require more tuning and validation before they are ready to be deployed [151, 160, 154, 155, 156].

The economics of OCR systems for low-resource languages are compounded by the absence of standard benchmarks, open-source toolkits, and community resources, and constrains the rate at which progress can be made and then achieved in collaborations [152, 153, 155]. This technological divide carries the risks of digital exclusion for speakers of these languages, so there is a need to address these problems for the overall global digital inclusion [155]

.

3.6.2 Historical Texts and Degraded Document Images

Another solid challenge area for modern OCR systems is processing degraded document images, historical materials, and texts with bad scanning quality – although each time progress is made in this area, it is dramatically better than the previous OCR system.

With modern OCR systems, advanced image quality enhancements and denoising techniques are applied to deal with quality issues. End-to-end deep learning models that combine denoising and character recognition, such as dual-output autoencoders are able to process noisy images and perform text recognition at once, resulting in a more accurate performance in comparison with traditional sequential mechanisms [161, 162]. Generative models such as modified CycleGANs and diffusion probabilistic models are employed to denoise degraded document images where paired clean and noisy training data is not available where some of them achieve over 60% improvement in word accuracy on real world noisy documents [163, 162].

Special lighting correction frameworks, including generative adversarial networks and dynamic light estimation, aim to alleviate the cases of uneven lighting conditions and shadow that may often deteriorate the quality of document capture [164]. Such improvements in preprocessing are necessary to keep the OCR performance in difficult capture contexts.

Strong segmentation methods, through object detection methods and graph-based algorithms, can even segment lines from extremely degraded documents with blurred text and bi-level background [165]. The combination of document layout analysis with image enhancement based on architectures such as Vision Grid Transformers leads to the accurate detection of text areas and increased recognition performance, mainly in complex historical documents [166].

Post-OCR correction based on large language models and encoder-decoder architecture can yield large performance boosts for historical documents with old orthography or complex layouts. Such advanced post-processing methods can achieve 25% relative error reduction compared to previous solutions [167, 168]. Systems for generating realistic degraded document images in a synthesized environment, based on GAN frameworks as well, to make OCR systems more generalize better on real world historical, and noisy documents led to a reduction in character error rates up to 11% [169].

3.6.3 Mathematical Formulas, Chemical Formulas, and Technical Diagrams

Mathematical formulae, chemical compounds, and technical diagrams cannot be handled easily by OCR systems for their special notations, their special writings and their two-dimensional appearances, which apparently are quite distinct from normal text.

Mathematical expressions show complex two-dimensional structures, different sizes of symbols, and complex logical relations, making them much more difficult for OCR systems than plain text reading. Such structural features are not well-handled by many traditional OCR methods, resulting in dramatically decreased accuracy rates ([170, 171, 118]). Although encoder-decoder models with Vision Transformers and sophisticated Transformer architectures have achieved high accuracy for printed mathematical formula recognition and fast generation of mathematically editable output [170], recognition accuracy is still sensitive to image resolution and quality, lower resolution image would significantly enhance uncertainty and error rates [172].

Convolutional neural networks have also proved very successful for the segmentation and recognition of handwritten equations, in particular for extracting and digitizing mathematical formulas from lined paper and educational documents [173]. But, mathematical notation is still a complex task for such systems.

There are unique challenges associated with chemical formulae, which are composed of symbols, subscripts, superscripts, and are formatted according to a domain-specific set of rules. As their notation is unique, off-the-shelf OCR models generally don't perform very well for chemical formulas [174]. Transformer-based models pre-trained on fine-tuned specialized chemical datasets achieve better performance which specially is observed with the model trained with smaller patch size (1x1) and with domain specific data augmentation techniques [174]. Multi-domain training regimes of both scientific and generic text show to be beneficial, although recognition of real-world artifacts and formatting conventions in all disciplines is still an issue.

Technical drawings are in need for advanced techniques which can perceive and analyze complex visual structures while text recognition may fail in this context. Convolutional Neural Networks can be taught to detect visual features in technical diagrams, automating analysis and evaluation in educational and professional contexts [175]. Advanced OCR models with layout analysis and location-guided transformers further enhance the recognition of complex documents with embedded diagrams, tables, and equations, that suffer from a high number of recognition mistakes and output repetition [176].

3.6.4 Processing in Real-Time and Analysis of Video Streams

Performing OCR on-the-fly for video streaming and live document capture remains a significant technical challenge in terms of speed, computational power and accuracy, particularly for dynamic scenarios. Classical CPU-based OCR processing adds significant delays, with mean latencies of around 48ms per image, making real-time processing of high volume video streams unfeasible [98]. With GPU-based solutions and streaming-optimized processing pipelines it is possible to achieve significantly lower latencies below 1ms per image, however, this requires specialized hardware and special-tailored software architectures [98]. It is challenging to provide a real-time OCR, particularly for deployment, parallelization and hardware acceleration lead to an overhead and an additional cost for the system.

Performance of OCR on real-time video processing is significantly influenced by frame

quality - high quality frames can result in recognition rates over 95% and degrading performance in bad lighting, blur of motion and low resolution [177]. Preprocessing techniques (such as thresholding, deviation correction, and noise removal) can enhance the OCR accuracy, but contribute to the overhead and complexity of the real-time processing pipeline [98]. Dynamic video content (differences in lighting, camera perspectives, motion, etc.) introduces extra difficulties in video-recognition quality throughout time.

Scaling real-time OCR for processing multiple video streams or high-volume live capture scenarios demands significant computational resources, especially if high precision and low latency are to be served at the same time [98]. Segmentation processes with Object detection methods such as YOLO can be faster and more accurate by concentrating the computation to relevant regions, but at the cost of added extensible processing layers that need to be optimised for real-time [177]. The trade-off between speed, accuracy, and computational cost is a primary bottleneck for a real-time OCR platform in practice [98, 177].

These challenges and opportunities underscore key areas for deeper research and innovation in OCR. Solving these issues would enable OCR technology to become more widely used in terms of language, document type and deployment, and would retain accuracy and performance levels ideal for practical uses.

Chapter 4

Problem Definition

Past chapters have illustrated the tremendous advancements in OCR systems, from classical methods to cutting-edge deep learning models, yet also exposed the severe shortcomings for domain-specific applications in sensitive sectors. This chapter introduces the key challenge of the creation of a legal OCR system built on AI under severe demands for privacy preservation and under edge computing constraints. We first describe the problem, and discuss the challenges it poses on legal multimedia processing given the complexity of document layout, the legal terminology and the need to perform processing on the client side to maintain data privacy. By reviewing the literature focusing on AI-based legal OCR, and identifying well-defined functional as well as nonfunctional requirements, this part defines the ground for the system design and implementation. The chapter is finished with specific use cases and success criteria intermediate results that will be used as guidelines for our solution development and evaluation.

4.1 Problem Statement

The legal sector is currently undergoing a major digital transformation due to the storm of documents and demand for fast, precise, secure processing solutions. Law professionals deal with bins of old documents including contracts, court filings, handwritten notes, and old case files that need to be turned into files they can search and analyze. Conventional manual processes to prepare a legal document have been used by counsel for decades, if not centuries, however such methods are becoming impractical in responding to a volume, complexity, and time sensitivity of today's legal workflow.

Automated processing of documents and OCR technologies present a lot of potential in this regard, as it allows for faster conversion of legal documents into machine-readable text, minimizing manual typing and review time, leading to better productivity" [178]. OCR systems convert scanned or handwritten legal records into searchable and analyzable data, enabling easy data access and advanced functionalities such as e-discovery and compliance checks [179, 180]. Additionally, automation reduces the chances of human errors during data extraction and in turn improve accuracy in legal processes and documentation [178, 181].

But the legal documents have some special features that make it pretty hard to be processed in the same manner as any general OCR service. Legal documents often consist of handwritten annotations, signs and varied formats, and so, accurate OCR is especially challenging, as handwritten OCR is less robust than printed text area recognition, especially for complex or bad handwriting [179, 180, 182]. Moreover, complex layouts, tables and multimodal content including text, images and signatures are commonly found in legal documents. These documents need advanced layout analysis and deep learning techniques to be detected and segmented properly [178, 183, 181, 184].

This complexity is enhanced by the quality and noise of scanned documents, which can cause text recognition to fail with a need for strong pre-processing and post-OCR correction to get usable results [185, 180, 183, 181]. There may be other challenges of text (history) recognition: Legal texts often contain several languages, (special) fonts, non-standard characters, this requires (OCR) system to be tolerant to such texts [183, 181, 186].

Apart from these technical limitations the legal space has tough security and privacy requirements that cloud-based OCR solutions cannot be used for confidential documents. Legal firms need to be regulatory compliant with laws such as the GDPR and also to preserve lawyer-client privilege and thus require on-premise or edge-processing solutions. This further adds to the challenges of computational power, efficiency and deployment constraints.

Thus, this research fills the gap and designs a dedicated Artificial Intelligence (AI)-based OCR system for the legal domain. The main purpose of this system is to produce extracted text from a collection of legal documents—ranging from contracts, court documents and handwritten notes—in a way that conserves the construction, structure and style of the extracted content to a certain degree. The development must be efficient on edge devices in order to provide privacy and security of data, but also maintain the accuracy and trustworthiness necessary for legal purposes.

4.2 Key Challenges in Legal Document Processing

Against this background, Section 2 presents that there is indeed a strong case for AI-based OCR system for the legal domain, however, the development and adoption of such systems are fraught with a multitude of technical and practical challenges that are particularly stringent within the legal setting.

4.2.1 Document and Content Complexity

Structural and content complexity of legal documents makes them a particularly challenging domain for OCR compared to regular OCR applications. Due to format varieties, complex layouts, and professional vocabulary used across legal documents, the recognition and extraction of TEXT in legal documents is a challenging task.

Diverse Formats: Legal document processing systems must be able to handle a wide variety of document formats such as typed contracts, handwritten annotations, scanned court filings, and multi-format documents where a document may include types of content that are mixed. The presented diversity poolings in Section 4.3 suggest that OCR systems need to deal with rich variability and that they should adapt processing per section considering features with which specifications of sentences in it are best captured. It is more challenging to handle content that involves a mixture of handwritten and machine-typed text due to the difference in nature of handwritten and machine-typed text where handwritten text recognition system is more difficult compared to machine-typed text, especially when both types are in use within the same document [179, 187, 188, 189]. The variation in handwriting style, quality of document and layout complexity, can greatly reduce recognition rates, with recognition accuracy usually behind that of typed text.

Complex Layouts: Legal documents are often presented in complex layout schemes, which defy straightforward text flow patterns, thus necessitating the use of state-of-the-art layout-specific segmentation for precise processing. Tables are one of the most difficult layout elements because to extract texts from tabular structure needs to perform layout analysis effectively for the row, column, and cell borders. For table structures, common OCR tools usually cannot deal with well, and therefore incorrectly aligned or scrambled text outputs [190, 191, 192, 189]. Signatory information is also a challenging constituent of the documents, the information is very ordered and regular and it is hard to find consistent rules along

with the natural variability and overlap between signatories and the rest of the document. Although deep learning procedures based on backbone networks such as ResNet-50 have demonstrated improvement in signature recognition and authentication, the combined signature extraction with the general OCR processing is still a challenging issue, especially because the system has to satisfy the high-pass for detection and the validation process [193, 189].

Specialized Terminology: Legal-related document text includes vocabulary that is specific to the field of study, which can have special formatting needs and even contain unique characters that OCR systems must handle with accuracy to interpret correctly. The legal language spiced with Latin terms, case citations, legal punctuation marks requires OCR engines specifically trained or adapted to the legal language. Preprocessing and layout analysis methods, such as image deskewing, partitioning, and reconstruction, have shown to potentially increase OCR accuracy up to 23% for documents with intricate layouts [192, 189]. Yet accurate recognition of specialized legal vocabulary is still a challenge that requires constant updating of OCR models and training corpora.

All these sources of complexities corresponding to the task require hybrid and various-modal models that can perform the semantic and visual understanding. It has been observed that the synergy among dedicated table, signature and handwriting processing modules results in higher extraction accuracies when supplemented by classical OCR techniques alone [190, 187, 189]. However, to obtain the reliability and accuracy levels required for legal applications, these sophisticated layout elements still demand special solutions and sophisticated preprocessing methods.

4.2.2 Security and Operational Constraints

With some of the most complex requirements for privacy and security in the history of professional industries, legal work can be especially challenging for AI-powered OCR systems. Personal assets of attorneys as well as their firm assets are contained in legal documents - there you will find highly sensitive information that is protected under privilege claims and a number of regulations that require data security.

Legal companies have to comply with a range of data protection laws such as GDPR, HIPAA and other region-specific privacy laws regarding the processing of sensitive information. These guidelines usually do not allow the sending of sensitive data to third-party cloud services, forcing the conclusion of on-premise or edge realization. The need for localized

processing creates limitations in terms of processing requirements as well as in system architecture and deployment planning that needs to be handled carefully in OCR system designs.

In order to resolve these strict conditions, some privacy-preserving machine learning methods show that they are suitable for edge-based document analysis. Federated learning allows edge devices to jointly learn models without exposing raw data, which greatly mitigates the risks to privacy [194, 195]. When used in conjunction with differential privacy — a technique that involves injecting deliberately calibrated noise into updates to a model — this helps further protect against inference of individual data, albeit with the need for careful tuning to prevent reducing model accuracy. Recent studies show enabling number of local updates as well as the tuning of differential privacy noise for maximizing the trade-off in the edge domain is feasible [194, 195].

Homomorphic encryption is another option that allows constant encryption of the data across each step of the analysis. This approach permits encrypted data computation while preserving private data end-to-end without decryption in the process [196, 197, 198]. Research has shown that homomorphic encryption, especially if combined with federated learning architectures, can also be used to enable a high level of privacy without relying on trusted third parties although it could introduce computational cost that should be controlled in edge-aware devices [196, 198].

More advanced PP methods use several different methods to achieve the highest possible privacy protection. Hybrid systems based on homomorphic encryption, secret sharing, and differential privacy appear promising towards full privacy assurance. These techniques can perform shallow training of model with homomorphic encryption on edge devices, deeper training operations using secret sharing protocols in cloud, and differential privacy for obfuscation against inference attacks [196, 195]. Furthermore, hybrid encryption along with Laplacian noise based differential privacy has been shown to result in better user anonymity and data confidentiality in edge-based federated learning systems [195].

These privacy-preserving practices add vital security properties, but they come with the performance cabinet of the friction that needs to be managed diligently. The computational overhead of homomorphic encryption and the possible loss of accuracy due to the differential privacy necessitate optimization with respect to the trade-off between security and operating efficiency. Legal OCR systems must strike a reasonable compromise doing so at the high levels of accuracy required by legal applications (where small errors can have huge conse-

quences).

4.3 State of the Art in AI-Powered Legal OCR

The advances in development of AI and DL have dramatically shaped optical character recognition capabilities, especially in niche domains such as legal document processing where accuracy, structure preservation, and privacy are of great importance.

4.3.1 End-to-End Document Analysis Pipelines

Recent studies in legal document digitization have seen a burgeoning trend towards integrated deep learning-based architectures where both text detection and recognition processes are embedded in one single pipeline. These end-to-end systems exploit convolutional neural networks (CNNs) for accurate text region localization and recurrent neural networks (RNNs) for sequential text generation and have shown to be effective for dealing with the challenging structural and linguistic information from legal documents [199]. The design of hybrid CNN-RNN is found to be highly effective in the legal domain where documents are composed with diversified layout, specialized terms, and varying text quality. It has been shown that CNN-based text detection models considerably outperform the rule-based methods in the legal sentence boundary detection task in terms of F1-score and computational cost [200].

Additionally, the incorporation of LSTM-based RNN architectures for text recognition has made it possible to extract reliably sequenced information from degraded or handwritten legal content and has shown high accuracy rates even for difficult document formats [201]. The utility of these end-to-end systems is not restricted to plain text extraction but they also retain the structure of the document and generalize well across the heterogeneous nature of legal texts, a desired capability for both high accuracy tasks and text formatting in legal tasks.

4.3.2 Layout-Aware and Structured Document Processing

The complexity of legal documents, due to the complex hierarchy, multi-column layout, and varied formatting, requires sophisticated layout-aware processing techniques that go beyond simple sequential text recognition as employed by conventional batch OCR systems. Pretrained transformer-based architectures have recently shown to be very effective for

processing structural legal documents, achieving significant improvements in precise layout analysis and accurate contextual text extraction compared to classical OCR approaches [202].

Recent transformer models tailored for document processing leverage spatial information by embedding layout features, such as token bounding box and positional encodings, into their input representations. This layout-aware strategy allows the models to capture the geometric and semantic connections between varied document components, achieving layout recognition accuracy scores of up to 97.3% on benchmark datasets [202, 203]. Then, Ackermann and coworkers proposed the LAMBERT (Layout-Aware Language Modeling for Information Extraction) model, which is designed to process visually rich documents while considering the spatial structure of the document during the process pipeline and was able to show better performance in extracting important information [204].

The incorporation of self-attention mechanisms in those transformer architectures allows them to capture longer-term dependencies across the sections in documents, hence achieving more accurate information extraction of contextually relevant information, even when legal terms and clauses are spread out over different pages or columns [205]. In addition, domain-specific fine-tuning of transformer models has achieved state-of-the-art results for legal text processing, with multi-level attentions trained to focus on the specific linguistic and terminological complexity that legal documents possess [206]. Such layout-aware transformer models show remarkably improved robustness to OCR errors and reciprocal ability to generalize on various types of documents and formatting characteristics that have been reported to be also a main limitation of the rule-based OCR systems when dealing with the complexity of legal document structures [207].

4.3.3 Solutions for Diverse and Difficult Text

For OCR technology, processing legal documents has to deal with a wide variety of textual contents such as handwritten marked addendums, signatures, multilingual paragraphs, and legal-specific terminology that demand task-specific processing strategies and exceed the coverage of the plain text recognition facilities. The significance of adaptive OCR solutions cannot be underestimated given such complexities without compromising the accuracy and reliability required in the legal arena.

Domain-specific adaptation and fine-tuning have shown good improvement for OCR performance of handwritten text in legal documents, including annotations and signatures,

both of which are key components of the legal document for the purpose of identification and interpretation [208]. The TransDocAnalyser framework is another testament to this trajectory, using encoder-decoder architectures with Faster-RCNN for region detection, Vision Transformers for feature extraction, and domain-specific decoders trained with legal vocabulary tokenizers to obtain state-of-the-art results on a variety of challenging legal datasets [208]. These domain-adapted models perform well at processing the mixed-content documents which typically involve handwritten as well as printed text, using the post-correction technique, proposed to handle recognition errors of legal terminologies, thus minimizing the domain-specific misinterpretation and improve the extracted handwritten information reliability [208].

End-to-end post-correction approaches, such as TrOCR with contextual language models, have also demonstrated strong domain-specific adaptation to handwriting styles and terminologies based on OCR systems and improved error handling, even when training data is scarce [209]. These hybrid models exploit visual as well as linguistic evidence for better recognition of handwritten legal annotations, signatures, and marginalia that play an important role in the context of complete legal form analysis.

The recent advances in deep learning frameworks for multilingual OCR in legal documents have pushed the boundaries of text authenticity preservation between multiple languages and scripts through intelligent neural architectures and transfer learning strategies [210]. The ADOCRNet architecture achieves state-of-the-art results by integrating Convolutional Neural Networks for feature extraction and Bidirectional Long Short-Term Memory networks for sequence modeling, trained using Connectionist Temporal Classification, with very low error rates on both printed and handwritten multilingual corpora [210].

Attention-based transfer learning has been shown to be especially effective for multilingual handwritten recognition systems and allows models to concentrate on important regions of the text [211]. These frameworks are designed to cater to various scripts such as Indic, Arabic, Urdu, or Latin scripts, through using huge multi-script training corpora and incorporating script identification modules as preprocesses to guarantee correct language identification and further processing [212, 213].

Pretrained transformer-based models, despite their computational cost, provide better OCR results for legal documents with a small multilingual dataset, and in the case of fine-tuning, they also allow adapting to different legal jargon and formatting between different

languages [214]. These state-of-the-art systems collectively tackle the core challenges behind the demanding layouts, mixed scripts and very high precision requirement expected in the processing of legal documents, whilst preserving the high level of text fidelity in a wide range of languages and scripts.

Document Segmentation

The document segmentation approach discriminates printed and handwritten text effectively in legal documents, and the OCR can be tailored to work on handwritten or printed text. Document segmentation categorizes each pixel of a document image into classes like handwritten text, printed text, or background and still generates a pixel-wise mask that separates different types of content as sharply as possible. This method is well suited for document processing, in particular for legal documents where handwritten annotations may co-exist with printed clauses and signatures side by side, requiring accurate separation in order to provide better recognition accuracy [215].

The recently proposed document segmentation models such as *DeepLabv3+*, which utilize CNN with atrous convolutions and spatial pyramid pooling to extract multi-scale context, have proven to be very effective in handling different handwriting styles and styles as in legal documents [215]. Mapmodels are learned from a corpus of labeled legal documents for which we know the corresponding positions of printed and handwritten areas and where they can therefore learn the visual differences between different classes of texts. During this segmentation stage, we pre-augment the document image to highlight the handwritten content (if contrast with the background is bad) and create a segmentation mask to assist OCR: printed script is processed with high-precision fonts recognition models, while handwritten script is addressed with specific models qualified for handwriting variation [215].

Semantic Segmentation for Handwritten and Printed Text Separation

Because the handwritten text is largely mixed with printed text in legal documents, the proposed approach can effectively segment the handwritten text out from the printed text, and thus targeted OCR could be used for each type to improve its recognition rate. In this pipeline, every pixel in a document image is labeled as belonging to classes such as handwritten, printed or background, creating a mask that identifies separated areas of content. Motivated by that of Malik et al. (2021), which leverages semantic segmentation for recog-

nizing rhetorical roles in legal documents, in adapting deep learning models to recognize the distinction between handwritten annotations, signatures, and printed clauses, with a particular focus on the challenges presented by legal documents containing mixed content [216].

The proposed approach uses the DeepLabv3+ model, which uses convolutional neural networks (CNNs) with atrous convolutions and spatial pyramid pooling to exploit context at multiple scales and handle the variability present in handwriting styles and printed characters typically found in legal documents [216, 217]. We train our models on an annotated corpus of text on legal documents where each hand-printed and printed text in various areas of the document is manually tagged, so it can learn to distinguish the visual and structural differences between hand-printed and printed text. Pre-processing such as contrast enhancement and noise reduction is performed to produce better accuracy of segmenting, particularly for damaged or scanned legal documents.

After producing a segmentation mask, the system sends the detected boxes off to dedicated OCR modules. Printed text areas are modeled using a high-quality OCR engine ABBYY FineReader optimized for high-accuracy recognition of printed text in the legal domain where it shows near-perfect accuracy for typed contracts and court filings [218, 219]. Handwritten text regions, such as annotations and inscriptions, are passed to a Vision Transformer-based model such as TransDocAnalyser, tailored for legal documents, for recognition where it is capable of handling varied handwritten styles due to its encoder-decoder design and domain-specific training. Such a disjoint manner ensures that each text type is analyzed by the most appropriate model, thereby greatly mitigating the cross-content confusion errors.

The document semantic segmentation provides accuracies of over 95% for legal documents containing mixed content, while the F1-scores for handwritten area detection are up to 93% over the reference datasets [216]. By decomposing handwritten and printed text, the method enhances OCR performance, and can also be used for help in more downstream tasks, such as document structure preservation and content indexing. In addition, lightweight versions of DeepLabv3+ can be run on edge devices like NVIDIA Jetson platforms.

4.3.4 Privacy-Preserving Edge Deployment

Data privacy and regulatory requirements are driving the development of privacy-preserving OCR frameworks tailored for deployment on edge devices, with the goal of confining sensitive legal data to local (edge) environments while preserving the high processing efficiency

and accuracy. This solution meets the precise need for accurately and securely processing confidential data in a manner that strictly adheres to restrictive data privacy laws and regulations like the GDPR and prevents documents from being processed in violation of court orders or professional rules.

Deploying OCR systems to edge devices with limited resources (e.g., smartphones) to process legal documents requires advanced model compression approaches which can reduce the computational overhead while preserving privacy. Quantization methods have been successful in reducing model storage and inference latency by quantizing model weights and activations from high precision floating point data to lower precision formats with minimal accuracy loss and remain compatible with privacy-preserving techniques such as homomorphic encryption [220, 221]. Complementary pruning methods alleviate this issue towards constructing a smaller neural network that does not sacrifice accuracy [222, 82], with adaptable pruning methods to tailor to device constraints, such that multi-constrained optimization towards accuracy, latency, memory, and energy can be traded off simultaneously.

Advanced compression methods involve privacy-preserving perturbation, applicable not only to compressing the model size but also actively securing private information in a way that makes parameter reconstruction harder. Autoencoder-based parameter compression paradigms are scalable regarding the degree of compression and provide communications efficiencies, and privacy protection, by distributing the parameter space during compression whilst adaptive compression schemes adjust the degree of compression based on access to device resources [223, 224]. The coupling of these compression schemes with federated learning and differential privacy techniques can additionally reinforce privacy preservation as it involves injecting controlled noise into model updates and the support of distributed training without leaking raw legal documents [225, 226].

The realization of privacy-preserving edge deployment in legal scenarios shows that sensitive court filings and legal documents can be executed offline, subject to severe regulations. These paradigms also allow legal tech companies to stay totally privacy-conscious, where all document processing happens on the edge (e.g., NVIDIA Jetson devices) and does not have network connectivity or can send data outside [227]. The aggregation of model compression techniques and secure computation methodologies, including homomorphic encryption for highly sensitive applications, offers a complete system to meet both the technical limitations of edge devices and the strict privacy demands of legal content processing workflows.

4.4 Requirements Analysis for the Proposed System

Fulfilling such special requirements for an OCR system, as in the digitization of legal documents, is indispensable for achieving an acceptable level of quality in transcript processing with high accuracy because of their complex design and layout, specific terminology used or privacy concerns, as well as for integration and fitting in with legal workflow processes.

4.4.1 Functional Requirements

An OCR system that is intended to digitally "read" complex legal documents will have to meet a number of critical functional requirements if it is to process the scanned pages efficiently, accurately, and reliably at a quality good enough for highly demanding legal document management and workflow integration [120].

The system needs to include advanced layout analysis techniques to reliably extract structural parts of different document layouts (text blocks, tables, multi-column layout, headers, footers, embedded images) without destroying the original hierarchy of legal documents [228, 192]. This need is particularly important for legal documents, which frequently include complicated formatting having sections, subsections, numbered paragraphs, edges, and citation structure that needs to be preserved for legal and referencing. A system should provide robustness against heterogeneous layouts, skewed document images, and different font types such as mixed fonts like proportion font (commonly used for legal documents) [192].

The OCR engine should provide accurate text recognition for text and images with a geometric layout [228, 120] which involve signatures, annotations, and marginal notes as important parts of legal records. This has implications for the support of legal terminology and complex language structures, thereby obtaining high precision when extracting domain-specific content such as case citations, statute references and legal precedents [229, 120]. The system must ensure accuracy for all types of documents – from the court filings, contracts, depositions and legal briefs that have so far been ingested to general correspondence and the like – regardless of the age of the document or subsequent quality degradation.

An essential set of pre-processing functionalities will include automated deskewing of the image, noise filtering, contrast enhancement, and quality enhancement from the tasks of scanned or photographed legal documents that can suffer from a variety of source quality concerns [192, 120]. These pre-processing steps are necessary to achieve the best possible

OCR results, particularly for historical legal documents, carbon copies, as well as documents that have been scanned multiple times.

The architecture should be able to carry out bulk digitization tasks described previously, but also perform those operations in real-time without sacrificing system performance and speed, thus it must include a scalable infrastructure able to cope with workloads that change dynamically, which is typical in the legal sector [120]. To enable seamless integration with today's document management systems, legal practice management software, and fast, secure data storage options, the e-filing system should be built on a modular microservices architecture [120].

Extracted text content needs to be converted by the OCR system to structured machine-readable content (JSON, XML, CSV etc.) so as to be integrated seamlessly with downstream legal workflow autosystems integration that handles publicly available documents such as court judgments, tribunal decisions, textbooks, legal search engines, and content management systems [120]. This structured output capacity is crucial for automating legal document analysis, content indexing, and interfacing with legal research databases and case management systems.

4.4.2 Non-Functional Requirements

The use and deployment of OCR technologies in the legal space requires meeting stringent non-functional requirements around quality assurance (litigation documents are of high quality and reliable), security (privacy compliance, NDA obligations), and operation (efficiency) standards and levels that are critical for the handling of sensitive legal documents and holding legal professionals accountable [230].

High-quality benchmarks for legal OCR are so important due to the fear of misreading critical legal text leading to high professional liability and legal judgments. The system must aim at achieving the lowest possible CER and WER, with the industry-leading systems showing CER $0.01. It must critically be robust and should work with acceptable performance for degraded or historical legal documents, even if CER on the ground remains higher [230, 231]. The system not only needs to perform well in terms of being accurate when it comes to the OCR text, but it must also be reliable and stable on a variety of documents, varied formats of those documents, and varied manners of processing those documents to be trusted in legal workflows where document integrity is of utmost importance [230].

Strong security features are a must-have for legitimate OCR systems of legal documents, as those are very sensitive and confidential. It should have end-to-end data security features, i.e., from encryption at the time of transmitting, encryption in stored data, and a multi-layered access control stretched across the various levels of access and roles, which restricts the document to be reviewed based on the user roles and the clearances [232]. To fulfill the obligations under applicable laws and regulations such as the GDPR, HIPAA, as applicable, and the industry-specific data privacy laws and regulations, the system shall provide real-time detailed audit trails, data retention policies, and user actions logs to monitor accountability and assist legal discovery processes[232].

The system needs to provide high-performance processing to meet the strict operational requirements of the legal domain, e.g., high-volume processing for low-latency processing to enable real-time or near-real-time legal operations such as court filing preparation and document review workflows [230, 232]. The requirements of scalability specify that the system can handle the ever-increasing workloads effectively without the performance degradation and peak processing requirements when getting ready for litigation in addition to the steady-state document managing tasks [232]. Optimal utilization of resources is especially important for edge deployment, where computational and storage resources can be limited and hence optimized algorithms that balance between speed of processing with accuracy, while minimizing hardware costs, is imperative [232].

The solution needs to be highly available with low downtime by providing redundant processing capabilities and automated failover to deliver uninterrupted services during time-sensitive legal events. Best practices for disaster recovery today include automated backup solutions and geographically diverse data storage, as well as fail-safe recovery procedures to prevent the loss of data in a legal practice environment where impeded or lost access to documents can have professional and financial consequences.

4.5 Use Case Scenario and Success Criteria

In order to verify the practicality and efficacy of the developed AI-driven OCR system for legal documents, a specific use-case scenario is depicted to illustrate how it is put into practical use and in order to set up quantifiable success criteria to evaluate how the system behaves in edge computing environment.

This introduction suitably connects the theoretical needs analysis from section 4.4 with the applied and the evaluation metrics that are to follow, all the while staying grounded in your thesis' fundamental concepts of legal document processing, AI-backed OCR, and edge computing deployment.

4.5.1 Use Case: On-Premise Legal Document Digitization

Real-world data demonstrates that locking OCR into infrastructure physically controlled by a law firm allows satisfying confidentiality and data residency requirements without diminishing speed or accuracy. Here's how Norton Rose LLP can now speed-chew-and-spit and yet respect tight data-residency rules: its networked multipurpose devices shove scans direct to an ABBYY Recognition Server farm in the London datacentre where conversion to searchable-PDF now takes a matter of seconds - and this in turn get sluiced straight to the Interwoven WorkSite DMS. The system is currently processing several hundred documents daily and is eating its way through a 12-million-page backlog while recording matter-codes at the scanner panel to ensure that every image has a defendable chain of custody for billing and audit purposes [219].

And with everything axis-ing in on centralized but self-managed infrastructure: "From there, the London finance department of Eversheds Sutherland migrated ABBYY FineReader Corporate to a Windows Terminal Server, converting a single operator, per-invoice bottleneck for supplier invoices into a portal-side, simultaneous process." Paper invoices were essentially batch-scanned, e-mailed TIFFs were delivered digitally, the same OCR engine is now called up by many clerks under a pooled license, to provide near-perfect accuracy, with almost no post-correction and a clean list of line-items that flow directly into the firm's accounting package via the cloud without a single document leaving the firm's network [218].

At the cutting edge of research, Naparstek et al. A minimalist (adversarial-example-trained) U-Net detector achieves F1 scores of 96.4 (FUNSD) and 95.9 (SROIE) running in 4.8 sec per page on a single Intel i7-5930 K core with just 2 GB RAM – almost 3× faster than CRAFT yet fully-local to commodity office hardware [233].

In addition to these detection developments, Garg et al.'s on-device multimodal classifier compresses an OCR-plus-vision pipeline into a model as small as 12-13 MB: the fused text-and-layout model achieves 89.8% accuracy on FOOD-101 with an evident 30% reduction in model size, demonstrating that full document understanding can exist entirely within a

smartphone sandbox—perfect for lawyers who need to triage files on the go without leaking privileged content to the cloud [234].

4.5.2 Success Criteria and Metrics

The performance of the proposed AI-based OCR method for the legal documents will be evaluated by using a multi-dimensional framework of quantitative and qualitative metrics, including accuracy and operational efficiency. The evaluation methodology defines primary metrics to assess text recognition accuracy, secondary metrics that measure system performance, and a strong evaluation pipeline to enable a fair assessment on a diverse set of legal document types.

The primary performance of the system will be evaluated by CER and WER, which are the most common metrics of OCR evaluation [235, 185]. CER represents the percentage of the characters erroneously recognized and is calculated as edit distance between the OCR output and the ground truth text, which makes it an essential tool for fine-grained accuracy evaluation which is especially important for complex legal documents with domain specificity—the use case of this study [229, 20]. WER is also used to report the ratio of misrecognized words, based on edit distance, which are particularly important to analyze how OCR errors impact document readability and subsequent legal processing tasks. [185, 235].

As legal documents are numerous and complexly formatted, an assessment will include Flexible Character Accuracy metrics – which take into account reading-order independence [229]. This blossoming metric is also flexible in comparing strings adjacent to each other (with zoning) and is not troubled by errors that come from OCR errors opposed to problems such as layouts or segmentation, which is crucial for legal documents including tables, signatures, and multiple columns [235]. According to standard targets provided in the OCR literature, the system will target a printed legal CER of $< 3\%$ and a handwritten annotation CER of $< 8\%$, with corresponding WER targets of $< 5\%$ for printed text and $< 12\%$ for handwritten text. These targets coincide with the rather strict accuracy requirements for the digitization of legal documents and also take into account the greater complexity that is brought about by legal terminology and document structures.

In addition to accuracy, the performance of this system will be evaluated in terms of document throughput (pages per minute) and latency (time to first result). Resource measurement statistics will be generated for CPU, GPU, and memory to track the resource consumption

on the target edge devices for hardware-bound optimal execution. Power consumption and thermal behaviors will also be measured to ensure continuous operation in a lawyer's office.

A large-scale screening pipeline will be set up using human-annotated ground truth of the text from curated legal document data sets [228, 235]. The evaluation framework will utilize the same tools for computing CER, WER, and similar metrics, overcoming potential deviance in the implementation and treatment of layout analysis. In cases where full ground truth annotation is costly, strategic sampling and partial annotation methods will be used to enable cost-efficient estimation of OCR accuracy to high statistical confidence [235].

Chapter 5

Proposed Lightweig Legal Document OCR Pipeline

The implementation of the AI-based OCR system for legal document processing, as detailed in this thesis, is at the proof-of-concept stage, aligning with the end-to-end system design and architecture proposed in Chapter 4. The implementation is guided by the intention of building a lightweight pipeline suitable to account for the challenges related to our problem definition:

- managing the layout of complex legal documents,
- working with printed and handwritten text, and
- providing a successful treatment for specialized legal terms, with the constraint of a one-man-army environment and low computational resources.

The implementation follows a modular design, relying on existing frameworks in order to develop a nine-step processing pipeline. Instead of training heavy deep learning models from scratch, we utilize proven OCR engines (ABBYY FineReader, TrOCR, Tesseract) within a multi-engine setup with intelligent pre-processing, segmentation, and post-processing steps. This strategy guarantees a practical and deployable system while preserving the accuracy and reliability needed for legal document work.

The chapter describes, step by step, the development of this implementation: from setting up the environment to finally integrating and testing the multiple components. Particular focus is given to the preprocessing pipeline tailor-made to handle even the most challenging legal content, the intelligent segmenter that divides between printed and handwritten, and the

advanced post-processing system that improves accuracy with multi-model consensus and legal dictionary validation. Such aspects as code frame, parameter settings, and optimization methods guarantee the implementation of the system effectively under limited resources, making it able to obtain the professional-grade performance for LD digitization.

It should be noted that this chapter presents a pipeline architecture that addresses the complete scope of legal document OCR challenges. While the full end-to-end system represents a comprehensive solution design, the implementation work has focused on developing and validating proof-of-concept implementations for critical pipeline components, specifically: (1) the multi-engine OCR coordination system with intelligent routing mechanisms, (2) the legal dictionary-based error correction module with Greek legal terminology, and (3) the multi-model voting system for consensus-based accuracy improvement. These core components demonstrate the feasibility and effectiveness of the proposed approach, providing a foundation for future full-scale implementation.

5.1 Development Environment and System Setup

The basis of a working OCR setup, for the most part, is a properly set up development environment that performs for both resources and features. The development environment is built to be lightweight, modular, and uses software and frameworks, many of which are open-architecture and several open-source in nature. Such setup is the result of the constraints of a single-student project with modest computational resources, yet enough computational resources to start virtually small scale development efforts on basic hardware platforms.

5.1.1 Hardware and Software Requirements

It is set up to run on low-end hardware configurations, being common on academic development setups. You will need at least 8 GB of RAM and a multi-core CPU (Intel i5 or AMD equivalent) for your computer, and about 50 GB of free disk space on your development machine for the development environment, model files, and test datasets. Although some parts of deep learning are better implemented on GPU than on CPU (as is the case for most deep learning software), the system is fully CPU capable and provides a CPU-only target for fast iteration without using specialized hardware.

The software stack is written in Python 3.8+ as it provides the largest ecosystem of com-

puter vision and machine learning libraries. The primary development environment utilizes a virtual environment-style isolation to provide consistency in dependencies and to be able to reproduce its deployment. The major system-level dependencies are OpenCV 4.5+ for image processing tasks, which should be built with a suitable system-level dependency for image codecs and GUI support. The ABBYY FineReader SDK implementation requires Windows or Linux support and license terms for educational applicability.

Package requirements for python are requirements-driven with core dependencies, such as: OpenCV for image processing, lxml for parsing ABBYY outputs (XML), fuzzywuzzy and python-Levenshtein for fuzzy string matching, and scikit-image for additional image analysis. The deep learning parts use PyTorch for DeepLabv3+ segmentation model and the transformers library to integrate TrOCR, with precisely version-pinned dependencies to achieve compatibility between the system parts. Memory usage considerations dictate that peak usage during document processing is for a large legal document, which is 2-4GB depending on the size of the document and the size of the processing batch.

5.1.2 Development Tools and Framework Selection

The development workflow brings a focus on code visibility and debugging features necessary for a complex multi-stage pipeline. As the main IDE, we use PyCharm Community Edition, which has strong debugging capabilities for the complex image processing and OCR interaction flows. The IDE configuration contains custom run configuration for different pipeline stages, thus making it easy to test and develop individual components without invoking the complete pipeline.

Version control uses Git with an organized branching model that isolates the development of individual pipeline stages. We organize this repository as follows: All functional components are organized into modules: pre-processing utilities at `src/preprocessing/`, OCR engine interfaces at `src/ocr_engines/`; post-processing modules at `src/postprocessing/`; shared utilities at `src/common/`. Configuration management uses YAML files to control parameter settings so that thresholds, input file paths, and processing options can be adjusted in a straightforward manner, without modifying any code.

The testing framework unifies unit and integration testing and is specifically designed for image processing workflows. The testing framework is primarily pytest and contains custom test images and mock OCR outputs as fixtures. Visual debug support comprises the integra-

tion with matplotlib to visualize intermediate results of the pipeline, with OpenCV to display images during the pipeline running live, and logging, which can output for you a very detailed log with the traces of each step across the corpus and may be important when you start debugging a really complex pipeline for document processing. The documentation follows the numPy docstring conventions and will, once finished, provide an easily digestible API documentation for the modules that will form the pipeline.

5.1.3 Implementation Scope and Validation Approach

While this chapter presents a complete pipeline architecture, the implementation approach has prioritized proof-of-concept development for the most critical and novel components. The multi-engine OCR system, legal dictionary validation, and consensus-based error correction represent the core innovations that demonstrate the viability of the proposed approach. These components have been implemented, tested, and validated to establish the foundation for the complete system, while other pipeline stages leverage established techniques and frameworks that can be integrated following standard implementation practices.

5.2 Document Preprocessing Pipeline

The preprocessing pipeline is also the heart of reliable OCR performance, especially in OCR when the high variation characteristics of scanned legal documents have to be dealt with. Legal documents bring many distinctive preprocessing challenges, such as old paper, different scan qualities, mixed sources, and variable lighting, which may have a large impact on the accuracy of downstream processing like OCR across documents. The four-stage preprocessing methodology deals with these problems systematically and improves the image quality on the way without losing textual content information, which is crucial when working with legal documents.

5.2.1 Noise Detection and Analysis

The noise detection module includes a noise analysis engine that provides full analysis techniques to distinguish and classify various forms of noise typically found in scanned legal documents. The processing starts with analysis based on histogram through OpenCV's `cv2.calcHist()` to look at the distribution of pixel intensities in the three-color channel.

Its statistical nature is used to distinguish between characteristic signatures of the Gaussian noise (broadened histogram peaks having large variance) and salt-and-pepper noise (isolated spikes at the extremes in the histogram).

The edge analyzer module uses Canny and Sobel edge analysis to separate real edges — those from a document — from noise-generated artifacts. The Canny edge detector is realized with `cv2.Canny()` with adaptive threshold, which produces highly-accurate localization of edges thereby enabling recognition of actual text boundaries. Sobel filtering in both directions using `cv2.Sobel()` applied in the 1st order on T,XY has the property of capturing the noise patterns in different orientations (up to 90°), and these may revolve around the text. The algorithm computes edge density metrics through analysis of the ratio of detected edges over image size; a higher than normal density of edges indicates noise which is then filtered aggressively.

The system performs variance analysis at multiple scales to detect local and global noise characteristics. Local variance computation is performed by a sliding window technique with adaptable kernel sizes (generally 5x5 and 11x11 pixels) to detect areas of unequal pixel distribution. A global variability assessment inspects the overall image statistics, whose variances are predetermined empirically by comparing clean and noisy legal documents. The noise detection output produces a holistic score that incorporates histogram analysis, edge density, and variance metrics, leading to quantitative recommendations on the choice of noise reduction parameters.

5.2.2 Image Enhancement and Noise Reduction

The noise reduction work uses a multi-method algorithm, which chooses different filtering policies according to the identified noise features in the detection stage. `cv2` for the Gaussian Blur is blurred = cv2.GaussianBlur() with well-selected kernel sizes and standard deviation values to tackle the global noise patterns while maintaining the text readability. Adaptive kernel size scaling depending on the estimated text-height is employed during the process with kernel sizes of 3x3 pixels for high-resolution scans and 7x7 pixels for less quality scans.

To remove salt-and-pepper noise, median filtering is applied via `cv2.medianBlur()` implementation with the choice of the kernel sizes depending on the quantized estimation of noise. The median filter size is computed with a progressive strategy: we start with 3x3 and

gradually increase to 5x5 and 7x7 for much aggressively noisy zones. The implementation filters the text with edge preservation to avoid strong filtering at the detected text edges, preserving the sharpest character while background noise is removed.

There is an adaptive parameter tuning that will analyze image features and optimize the filter parameters based on the filter parameters of each document. The algorithm computes local image statistics based on the mean, standard deviation, and edge density to adaptively adjust the sigma values for the Gaussian blur operation and the kernel size for the median filter. Quality measures evaluate pre-and post-filtering results in terms of signal-to-noise ratio and edge-preservation score. The noise reduction pipeline is armed with validation logic to prevent over filtering by monitoring text region contrast and falling back to more conservative parameters if character degradation is observed through auto quality checks.

5.2.3 Binarization and Contrast Enhancement

The binarization method offers several different adaptive thresholding techniques that can be used in scanned legal documents with different lighting conditions. Otsu's method, implemented by `cv2.threshold()` and `cv2.THRESH_OTSU` flag, generates globally the optimal threshold and generally works better on documents (or similar) if they are bi-modal. Preprocessing histogram analysis is incorporated to verify bimodal assumptions and the method automatically degrades to an adaptive approach when global thresholding fails.

Adaptive Thresholding is through `cv2.adaptiveThreshold()` using mean as well as Gaussian weighting methods for local illumination variations. It uses the Sauvola method via custom realization that computes mean and standard deviation within a window of adjustable size. Selection of size p of the neighborhood is based on estimated character sizes, typically p=15x15 pixels for standard text to p=31x31 pixels for larger legal document headers. The k-parameter in Sauvola is dynamically computed by analyzing local contrast and it can vary between 0.2 for high contrast to 0.5 for low contrast areas.

Local neighborhood-based processing applies a sliding window analysis, which partitions the documents into overlapping sections for autonomous threshold estimation. It applies gaussian-weighted averaging across neighbor points to suppress the boundary effects and produce smooth threshold variations from region to region. Illumination variation is treated using gradient-based detection and correction algorithms that detect and correct systematic brightness variations among document pages. The binarization process ends with morpho-

logical operations via `cv2.morphologyEx()` to remove contour noise pixels and small gaps in character strokes, and how we determine the structuring element by estimating the character stroke width analysis.

5.2.4 Skew Detection and Correction

This skew correction code uses Hough Line Transform to scan for such documents and fix their orientation. The Hough Line Transform is a feature extraction technique in computer vision used to detect straight lines in images. It works by transforming points in the image space (x,y coordinates) into a parameter space called Hough space, where each line can be represented by two parameters. The process starts with an edge image generated with reasonable Canny parameters for text documents, then cv2.HoughLines() analysis in the previous step to recognize prominent linear structures pertaining to text baselines and document borders.

Angle detection algorithms examine the distribution of detected lines to determine the dominant direction of the document. In implementation, histograms of line angles are computed, usually accumulated at 0.5-degree intervals, and peaks are located which correspond to text line orientations. Large rotation angle estimation uses median filtering of detected angles and outliers detection to process documents with mixed orientation or more than one block of text. The system verifies the detected skew angles based on consistency by comparing the orientations of the lines in various parts of the document.

Computing a rotation matrix with cv2.getRotationMatrix2D() produces the correct transformation matrices for skew correction. This implementation enables automatic boundary processing, such that the size of the output image changes in response to the rotated content not being cropped. The two-level interpolation method selection is done via bicubic interpolation to achieve high-quality interpolated results with a moderate computational burden suitable for the resource-constrained scenario.

Aligning the text validation applies post-correction quality checking based on horizontal projection analysis. It computes row-wise projection amounts of the pixel densities and measures the regularity of the text line spacing to check whether the skew is corrected well. Quality metrics are variance of text line positions and consistency of inter-line spacing, and a re-processing stage is applied with re-defined parameters if alignment quality does not meet a pre-set acceptable level. The skew correction pipeline ends with geometric verification to en-

sure the corrected documents have correct aspect ratios and text proportions consistent with later stages of OCR processing.

5.3 Document Segmentation System

The document segmentation system is an important module to meet one of the main legal document processing challenges, which consists of accurately recognizing and segmenting printed text from handwritten annotations, signatures, or notes. It is common to encounter heterogeneous document structure in legal documents that need different OCR methodologies to achieve desirable accuracy. The semantic segmentation-based implementation generates accurate region masks that form the basis for subsequent OCR engine selection, ensuring that all types of content are processed with the best matching recognition model.

5.3.1 Printed vs. Handwritten Text Classification

The classifier is based on a custom DeepLabv3+ architecture tailored to the binary segmentation objective of recognizing printed versus handwritten text in legal documents. We use a pre-trained ResNet-50 backbone from PyTorch's torchvision library and modify the original classification layers to predict two-class segmentation masks for printed and handwritten regions. The process of model adaptation is to freeze the initial convolutional layers to maintain their general feature extraction capability while we fine-tune the deeper layers and the segmentation head for the characteristics of legal document content.

Training data preparation applies to the scarcity of labeled legal documents by a two-fold strategy, including synthetic data generation and carefully annotating samples of legal documents. A contrastive loss applied on augmented input examples simulates different scanning conditions and paper aging effects, as well as ink variations in the text, providing the model robustness. Annotation tools use LabelMe incorporation for accurate mask generation at the pixel level, especially focusing on boundaries between printed and handwritten text. The training data set consists of a wide range of data types such as contracts with handwritten changes, court forms with filled fields, and notarized documents with signatures and stamps.

Light model realization aims to be accessible to run on edge devices and computational power on the device, ensuring that the segmentation accuracy is sufficient for the resource-limited scenario. The model structure uses depthwise separable convolutions in the decoder

part to decrease the number of parameters and the complexity. Relevant prior art for compressing models includes post-training quantization using PyTorch's quantization toolkit, where the model size is approximately reduced by 75%, and segmentation accuracy is unchanged within 2%-3% of that of the full precision model. This is implemented by dynamic model loading, where whether a full precision or a quantized model should be used is automatically determined according to the available hardware.

5.3.2 Zone-Based Document Analysis

ROI extraction computes connected component analysis that extracts discrete text regions and structural elements from the segmented document. The code leverages cv2.findContours() with hierarchical contour detection to extract text regions and maintain spatial relationships between various sections in the document. Contour filter is used to suppress noise artifacts but preserve text regions by applying an area-based threshold and aspect-ratio constraint. The software has built-in support for table rendering, signature blocks, and multi-column documents typical in legal documentation.

Mask generation: Designing a mask generation module would allow for harvesting structured output while enabling pixel-level classification along with the semantics of the document layout. The approach produces a series of map layers such as binary printed/handwritten classification to make recognition, confidence maps for indicating segmentation certainty and structural element identification. Post-processing of masks uses morphological techniques to remove border artifacts and maintain uniformity between region definitions. The system comprises mask validation logic that looks for plausible size distributions and spatial coherence of labeled regions.

Validation of the segmentation quality results with a set of automated quality assessment metrics, which measure the accuracy of the segmentation without relying on ground truth annotations. The process scores the coherence of the regions based on the consistency of local texture within the classified regions, the quality of the sharpness of boundaries (e.g., printed/handwriting), and a logical arrangement of material types based on their spatial distribution. Confidence scoring summarizes pixel-level classification confidence into region-level reliability measures which can be used by downstream processing to focus on reliable regions and mark the uncertain regions for manual review purposes.

The zone analysis algorithm has geometric feature extraction and assigns each region

statistical descriptors from these geometric features. Features consist of bounding box size, rotation angle, aspect ratio, and their relative location in the document structure. These geometric marks help the OCR engine make routing decisions and enable us to extract document structure metadata. The use of a region hierarchy that reflects the logical document structure allows the document to be reconstituted correctly once any textual annotations, formatted blocks, or annotation regions have been added, modified, or deleted.

5.4 Multi-Engine OCR Processing System

The multi-engine OCR system performs text recognition based on the coordinated execution of multiple OCR engines, each designed for a specific type of content as determined during the segmentation phase. This strategy results from the synergy of different OCR tools: ABBYY FineReader excels in processing images of high-quality printed text, TrOCR is able to process images including handwritten text content, and Tesseract performs OCR on specific types of images, including some especially difficult cases. The system's architecture guarantees an integrated handling of the engines and that the output formatting and the extraction of metadata remain constant along all processing paths. This multi-engine approach has been implemented and validated through proof-of-concept testing.

5.4.1 OCR Engine Selection and Configuration

The integration of the ABBYY FineReader SDK is carried through a detailed interface, which can fully exploit the possibilities of the engine for legal text recognition of printed text. We use the Python API for the ABBYY SDK to define processing parameters tailored for pertinent characteristics of legal documents. Text recognition settings consist of language model configuration on legal texts, special dictionaries (legal names and Latin expressions), and preprocessing functions that enhance the image before the pipeline's initial stage of image improvement. The ABBYY configuration uses automatic page orientation detection turned off to use the pipeline's own skew correction, custom confidence threshold settings that result in providing character-level confidence scores, and XML output format that saves extended spatial and formatting information needed for the reconstruction of the structure of legal documents.

The TrOCR follows the Microsoft transformer-based handwritten text recognition model,

which is available in Hugging Face's transformers library. At inference time, the implementation fine-tunes the pre-trained TrOCR on annotated legal handwriting samples, when available, or uses learned prompt engineering and post-process adaptation. Model loading relies on the efficient memory representation of the transformer architecture in PyTorch, utilizing the solution's model quantization and attention optimization functionality. The TrOCR integration has custom pre-processing to dynamically adapt handwritten regions to the input format expected by the model, consistent image resizing and normalization as well as padding strategies that maintain handwriting features and fit model requirements.

Tesseract integration acts as a fallback and as a validation, performed through the pytesseract wrapper with a large set of configuration optimizations. It uses Tesseract's page segmentation modes (PSM) designed for different regions of the documents as PSM 6 for uniform text blocks, PSM 7 for single text lines, and PSM 8 for single words in signature blocks. A language model's configuration comprises their training on custom data (in case an institute provides legal texts) for the presence of legal terms, the whitelist/blacklist of character sets particular to a legal document style contents, etc. The Tesseract layout allows confidence score extraction and detailed word-level positioning information providing for direct comparison with other engine outputs.

5.4.2 Intelligent Routing Mechanism

Routing logic based on segmentation decision tree: For example, which bounding boxes to which engine (core back-end) - Robust decision tree for assigning a document's regions to OCR engine based on results of classifier, confidence scores, and region properties: ratio, area, height, etc. The technology inspects the results of the segmentation step to identify what kinds of regions it has found — printed text regions are automatically sent to ABBYY FineReader with known high accuracy; handwritten regions are routed to TrOCR for specialized processing; mixed or unsure regions are sent to multiple engines so that the extracted content can be compared. The routing system also has spatial analysis to take into account the size of the region (width and height), aspect ratio, and region location in the document structure for engine selection.

Zone-specific OCR engine association uses metadata-based processing, which keeps a detailed record of which engine processes each region of the document. In performance, the implementation generates processing manifests that specify engine schedules, processing

settings, and the desired output formats for each region. Such metadata allows complex post-processing flows that might perform error correction and confidence elaboration depending on the engine used. The system includes load balancing logic that distributes processing across available engines when multiple regions need the same OCR in a particular language, ensuring the fastest processing time while not sacrificing result quality.

Metadata extraction and coordination system adopts a common data model (CDM) to normalize the output of multiple OCR engines into interchangeable formats that downstream processing can consume. It provides common schemas across text, geometry, confidence values, and formatting which adapt to the range of output formats of the various engines. Normalization of the coordinate system ensures that all spatial information is in the same document coordinate system regardless of which processing engine you are using. The coordination system has support for timestamp tracking, processing duration measurement, and resource monitoring, which enables us to analyze and optimize the multi-engine solution performance.

5.4.3 OCR Output Processing and Standardization

Creating a consistent output format solves the problem of diverse output, allowing for the differences in the structure and the content of the output of the various OCR engines. The concrete specification includes a complete JSON schema for textual content, spatial location, confidence scores, and more, all in a consistent format. Parsing of ABBYY XML output is parsed using lxml to retrieve detailed character-level information to the degree of confidence, the font, and the position of characters. All predictions extracted from the transformer model are consolidated in TrOCR output processing into a standardized format, with the confidence being represented by the attention weights. Tesseract output standardization extracts word-level confidences and bounding boxes from the engine's TSV output format.

Normalization of coordinate systems applies geometric transformations such that all spatial information is referenced at standardized document coordinates independent of input preprocessing or engine-specific coordinate systems. The extraction is stored as inverse transformation matrices of all geometrical operations done in preprocessing in order to reverse transform the coordinates extracted from OCR to original document coordinates. Such normalization is very important, especially in legal documents where specific spatial relationships between text elements, signatures, and structuring elements are required for legal purposes

and for document reconstruction.

The process extracts and processes confidence scores in a manner that provides a common confidence framework for comparing confidence evaluations of different engines. First, it normalizes whatever confidence levels various OCR uses so that it's all on the same 0-1 scale from ABBYY's character level confidences through TrOCR's attention scores to Tesseract word level confidence. Statistical characterization of the individual engine confidence distributions is used to calibrate confidence thresholds so that the reliability indication is consistently interpreted across the multi-engine system. The confidence processing features confidence propagation algorithms that compute region-level and document-level confidence scores from character and word-level confidence measures, thereby providing hierarchical confidence feedback that is useful for quality control and for prioritizing manual review.

5.5 Post-Processing and Quality Enhancement

The overall post-processing pipeline is the final stage at which the raw OCR results are processed to become accurate, reliable text with the help of complex error detection, correction, and quality validation methods. This step is especially important for legal document analysis, where high precision is required and legal jargon has to be accurately identified. The method takes advantage of a three-phase twin straddled approach incorporating multi-model consensus, domain-specific legal dictionary validation, and confidence-based quality assessor to meet the stringent accuracy requirements in document digitization for legal applications.

5.5.1 Multi-Model Voting System

In the voting system of this type, multiple OCR engines are exploited to combine the strengths of their complementary recognition results to compensate for recognition failures. Levenshtein distance-based comparison is the heart of the consensus mechanism, utilizing the python-Levenshtein library to compute edit distances between matching segments of text to the set of engines. The method splits OCR outputs into comparable units (usually words or short phrases) and calculates Levenshtein distances between them over pairwise combinations to match up regions of agreement and dispute. Distance normalization penalizes the variation in length between observations so that the distance values are normalized as the

edit distance divided by the maximum length of the strings so that the comparison metrics across different strings are consistent. Also, it has been prototyped and tested to demonstrate its effectiveness.

Common OCR error pattern recognition adopts a unified error taxonomy obtained from the empirical study of OCR engine behavior on law documents. The framework incorporates error pattern databases, which record the most common misrecognitions experienced by the individual engines: (i) ABBYY's difficulty distinguishing between similar looking Greek characters such as "ι" and "1", (ii) TrOCR's confusion of some connected cursive letters, and (iii) Tesseract's high sensitivity to image quality. Pattern detection for errors is done using regular expressions matching and character substitutions to find likely OCR errors. The implementation incorporates context-aware error detection, which takes the nearby context of the text making sure the spelling check happens with the right kind of possibilities, which is more important in legal vocabulary where character level precision is of more importance.

The confidence-weighted voting method is a type of advanced decision-making model which can take into account inter-engine consensus and per-engine confidence scores. A plurality vote model incorporates the resemblance score in the voting, but allows giving more or less weight to the decision of any particular engine by other scores like confidence scores per engine (specific for each engine), historical accuracy level using similar text regions, and the degree of agreement with the voted decision by other engines. The algorithms rely on a variation of the Borda count algorithm, such that higher confident engines rank higher than lower confident engines which have a higher voting weight. Tie-breaking logic was applied contextually, favoring engines that show better performance for certain displacements (printed vs. handwritten) or types of content (e.g., different document regions). The voting system consists of decision-making based on thresholds, where a minimum degree of confidence is asked for a decision to be accepted and where uncertain regions are tagged to be manually reviewed or to be prolonged to cover more and then reconsider.

5.5.2 Legal Dictionary-Based Error Correction

Legal term dictionary integration: For OCR error correction in legal documents, the software applies a specialized Greek legal dictionary with legal terminology. The dictionary includes corporate entities, terms in use in contracts, court language, and Latin expressions used in Greece as well as the Latin terminology adopted from Greek law. Dictionary com-

pilation is based on official legal resources, such as authoritative court documents, standard-form contracts, and legal reference works. The implementation preserves a distinct hierarchy of dictionaries from highly confident exact matches of important legal terms, fuzzy match candidates of possibilities, to context-validating content requiring surrounding text analyses. This dictionary integration approach has been implemented and validated using a curated Greek legal terminology database.

Fuzzy matching functionality is implemented using the fuzzywuzzy library with tailored scoring algorithms that are fine-tuned for the properties of legal terminology. The system made use of various matching strategies such as simple ratio comparison to get an initial similarity score, partial ratio matching for correct subsequence types, and tokens-based matching to tackle the variations of words. The minimum level of similarity is dynamic and varies with term criticality; higher similarity (>95%) is needed to match legal entity names and contract-critical terms, and lower similarity (>85%) is allowed for common legal vocabulary. This process also involves phonetic matching, using phonetic algorithms specifically developed for the Greek language, and taking into consideration known speech-based OCR errors.

By integrating SymSpell, users get state-of-the-art spell correction suited to the legal domain language. The system adapts SymSpell's dictionary with legal terminology and sets edit distance parameters suitable for OCR error behavior. The legal terms prioritization guarantees domain-related corrections to be proposed rather than general word suggestions. The system provides support for compound words, considering the complicated compound word structure of the Greek legal terminology. Contextual validation utilizes n-gram analysis to ensure that suggested corrections are also semantically correct under the context of legal documents and can avoid inappropriate corrections that change the meaning of the law.

5.5.3 Confidence Score Analysis and Validation

ABBYY XML Parser is based on lxml and is able to extract detailed confidence information from ABBYY FineReader's XML output format. Character-level confidence scores in the XML are used by the parser to collapse this to form word, line, and paragraph-level confidence statistics. Character parameters extracted are font size, style information, and positioning which belong to the broader context of confidence interpretation. The module takes XML schema changes for various ABBYY versions into account, but is able to output OL-versioned old formats as well.

By having character-level confidence extraction, it generates very granular values of reliability, so we can have a very precise prediction on which part of the text gives high uncertainty. The process maps ABBYY's confidence values (usually between 0-255) into a normalized score that can be compared with the other confidence numbers of the OCR engines. Confidence distributions are analyzed statistically and patterns corresponding to categories of OCR errors are determined to allow quality prediction. The system further includes confidence calibration using empirical validation against ground truth data or by converting raw confidence scores to probabilities of overall accuracy.

Adaptive filtering dynamic quality assessment applies threshold-based quality assessment where text regions can be automatically determined whether to be subjected to further manual review/reprocessing. It is based on thresholding at various levels, such as minimum acceptable confidence for automatic acceptance of the reaction, moderate thresholds at which additional validation is triggered and stringent thresholds that will require manual verification. Quality flags are derived from a confidence analysis, whereby the type of quality problem is indicated by the specific type of flag raised: low overall confidence, high confidence variance in text regions, or confidence patterns related to well-known OCR failure modes. The review package has quality score batching aggregation for document-level quality metrics and batch-process-style quality control reporting.

The confidence analysis pipeline is completed with uncertainty quantification that furnishes statistical measures of the reliability of the results. The application computes confidence intervals on the estimates of OCR accuracy so that legal professionals can assess the extent to which OCR can be statistically reliable on digitized text. High-quality reporting provides rich, nuanced assessments that emphasize high-confidence accurate regions and uncertain areas needing further validation, so as to enable principled decision making about whether OCR output is in fact fit for use (legally speaking).

5.6 Pipeline Integration and Data Flow

The pipeline integration engine ensures that all the stages execute in concert to feed data efficiently from raw document input to final validated, well-formed text output. This layout manager handles the complex interdependencies of the image pre-processing, segmentation, OCR processing, and post-processing phases while ensuring data integrity, error processing,

and performance tuning at all levels of the system. The real implementation of our system is focused on being modular and fault-tolerant as well, allowing it to work properly even if some of the components are not able to process the information for some reason (e.g., too much pressure on the system, lack of availability of resources, etc.).

5.6.1 End-to-End Pipeline Architecture

The sequential processing pipeline has a modular stage composition; each processing template is designed to function as a standalone unit linked by clearly defined interfaces for the input and output data. The implementation employs a pipeline mechanism where the document data is passed through a series of transformation stages, and at each of the stages metadata and result information is added to the data structure. The process starts with document ingestion that checks input formats, does initial quality assessment, and produces processing manifestations that record the document metadata at various stages of the pipeline. Stage coordination uses a controller that receives all logic of stages interrelation, progress monitoring, stage execution order, and dependencies between stages.

The data passing mechanisms provide efficient serialization and deserialization and thereby reduce memory overhead, whilst retaining processing state between stages. It uses the pickle protocol in Python that allows for storage of image arrays and coordinate mappings with nested metadata in complex data structures alongside memory compression for large documents. Intermediate result caching uses a hierarchical storage method for the processing output result at important pipeline checkpoints in which selective reprocessing (of a subset of stages) is possible without running the whole pipeline again. The apparatus also has data validation at every transition between states to ensure that the data stored is complete and consistent before forward processing occurs.

Error handling and recovery mechanism supports detailed trapping and degrading of processing capabilities when individual units fail. Such a multi-level error handling is achieved by catching the exceptions at the component level, asking the stages to perform fallback steps in case of errors, and asking the pipeline to perform recovery. Rigorous error handling takes care of out-of-memory scenarios, timed-out processing, and corrupt input data via exponential back-off pattern retrying. It also offers partial processing functionality, which is able to complete the execution of the pipeline even when some modules/steps fail generating comprehensive error reports that detail which components have failed and what solving strategy

may be used.

5.6.2 Performance Optimization for Limited Resources

Memory management techniques cope with the computational and memory-intensive nature of high-resolution legal document images on several processing levels. Lazy loading has been utilized, allowing image data to be held in memory only if needed for active processing, with automatic garbage collection to immediately reclaim memory at the end of a stage. Image compression comprises dynamic resolution scaling to scale down image dimensions at processing stages that do not require full resolution, and are then scaled up for OCR processing that uses high-quality input. Memory monitoring, which monitors the amount of memory, and relieves memory pressure by temporarily storing files when there is not enough memory and the used RAM is about to reach the system limit.

Batch processing is optimized by adaptive batching strategies that adapt processing parameters according to the system's resources and the properties of the documents. The service calculates optimal batch configurations based on document complexity metrics like file size, estimated text density, and degree of throughput processing required. Batch processing logic groups together similar documents to process them most efficiently, with different types and complexity levels of documents being sent to various processing queues. The system incorporates dynamic adaptation of batch size that downsizes batches under memory pressure and upscales batching when resources are available.

It is important to define an elaborate storage hierarchy to achieve a balance between processing speed and storage presence for caching intermediate results. The implementation has multiple levels of caching, including in-memory caches (for frequently accessed data), disk-based caches (for intermediate processing outputs), and persistent caching stores (for expensive computations, such as the results of model inferences). Cache management uses LRU (Least Recently Used) eviction with size bound to avoid the growth of the cache demanding terabytes of storage. The system further includes cache invalidation logic that refreshes cached results on the fly when processing parameters are updated or when cached information is outdated.

5.6.3 Quality Control and Validation Framework

Automatic quality control applies extensive validation steps to monitor processing quality at each processing stage without human intervention. The proposed system contains a post-processing image quality evaluation which verifies the effectiveness of noise removal, quality of binarization, and accuracy of skew correction using statistical measures. Quality validation of segmentation is performed using automatic scoring by considering the coherence of regions, the accuracy of the boundaries, and the confidence of classification. OCR quality control looks at character-level confidence distributions, text coherency metrics, and cross-engine agreement to unveil potential accuracy problems.

Manual review outlet points are structured interfaces for human oversight at important decision points within the pipeline. The system detects review triggering conditions such as low confidence scores, high cross-engine disagreement, and abnormally looking documents that need cross-examiner judgment. Review interfaces show original documents side by side with the processing results, which outline areas of ambiguity and offer features to manually correct the content. The system has review workflow management functionality to monitor manual interventions, log and record audit trails, and introduce human feedback into quality improvement.

Monitoring and logging performance have rich built-in instrumentation for the efficiency of processing, resource usage, latency within every component of a pipeline, and quality of data. The decoder includes detailed timing logs, which are used to identify processing bottlenecks and optimization possibilities. Resource monitoring is concerned with resource usage: CPU utilization, memory consumption, and disk I/O patterns to detect resource limitations and to support capacity planning decisions. Through quality metric logging, accuracy, confidence scores, and error patterns are collected and tabulated to facilitate continuous improvement. The monitoring system has triggers to inform operators about processing failure, performance loss, or quality attention.

The validation framework is completed by Decoding Results into ready-made summaries of processing, quality levels, performance tracking, and reporting for both technical monitoring and economic reporting. Reports accumulate document-level processing statistics, summaries of batch processing, and trend information in which we start to observe patterns relating to processing quality and efficiency over time.

5.7 User Interface and System Integration

The UI and system integration elements represent the indispensable link between the high-tech OCR processing pipeline and the everyday requirements of legal professionals in need of simple, reliable access to document digitization capabilities. In view of the resource limitations of this single-student implementation, the interface design emphasizes functionality and robustness over advanced features, for law practice materials core workflows that facilitate efficient legal data processing with the security and quality requirements typical of applications intended for legal treatment.

5.7.1 Output Generation and Export

With structured text output formats, a number of different types of content can be provided to meet the differing needs of those who need document content, such as legal professionals who may need information in different formats for different uses. The JSON export option allows the representation of hierarchical documents which also include spatial relationships, confidence scores (CS), and processing metadata with the original contents of the text being extracted. The JSON schema contains document-level metadata (e.g., processing timestamps, engine configurations, quality assessments), page-level details (e.g., layout analysis, region classifications, and text-level annotations with confidence scores and spatial coordinates). XML export can produce structured output for legal DMS systems, which require XML data interchange.

Confidence report production provides in-depth evaluations of OCR accuracy and reliability, helping lawyers determine whether to rely upon the document. The application produces full reports about confidence, including statistical summaries of confidence score distributions, detailed region-by-region confidence analysis showing where the uncertainty lies, and comparative confidence analysis when several OCR engines are run on the same document. Visualization of confidence maps is given as easily interpretable colorized overlays on original document images. These confidence thresholds provide users with a mechanism to pinpoint text that possesses desired reliability for various legal applications, enabling users to customize a standard or level of reliability.

A list of errors, summaries, and statistics compilation for the analysis of systematic processing errors/quality issues in need of attention. The system produces error reports that clas-

sify various types of processing errors such as OCR engine failures, low confidence regions, and dictionary correction recommendations. Statistical summaries comprise error frequency distributions, frequent error patterns, and processing rates for different document types. It has processing logs to follow errors and the process quality development over time and to improve parameters and methods permanently.

5.8 Testing and Validation Framework

It is used to define quality assurance measures to test and validate the OCR pipeline to be reliable, accurate, and robust under normal to extreme operating conditions. In light of the importance of legal document processing, the testing strategy includes various levels of validation ranging from unit testing, through component verification and system testing up to end-to-end system testing. Both the functional correctness and performance characteristics of the framework are addressed, as well as systematic methods are presented for recognizing and rectifying processing problems which might arise and can affect the accuracy of legal documents.

5.8.1 Unit Testing Implementation

Strategies of individual components implement extensive test cases for each pipeline module with pytest as the main testing framework. The system generates sealed testing environments of preprocessing modules and their testing cases, which include tests of noise detection, enhancement, binarization, and removal of skews. Each preprocessing function is tested using synthetic test images that emulate different noise levels, illumination changes, and other possible document conditions, with its own test coverage. Segmentation component testing is conducted using ground truth through automatic comparison of umbra classification accuracy, region extraction precision, and boundary detection quality to mock segmentation masks.

The generation of mock data generates comprehensive test data sets emulating real-world legal document properties without relying on actual sensitive legal materials. The introduction of synthetic legal documents with controlled features such as the density of text, the presence of both printed and handwritten information, as well as varying quality of paper and reproduction artifacts. Synthetic OCR output generation produces synthetic test data for

post-processing modules, with known and controlled confidences, error rates, and accuracies, and allows thorough testing of error correction algorithms. The mock data system has production of parameterized tests that spawn thousands of test cases uniformly generated from across varying input situations and underlying handling parameters.

Automatic execution of test cases enforces Continuous Integration practices to ensure code changes do not result in regressions or degradation of performance. The implementation uses pytest fixtures for test data and processing environment management, and the parameters of tests encouraged variation over various input conditions to validate parts of the system. Test automation performance regression testing observes how fast and how much memory it takes to run the software; to detect how much it (de)optimizes. Test coverage analysis, by using pytest-cov to track and achieve full test coverage of important components of the pipeline and to uncover untested code paths that may contain bugs.

5.8.2 Integration Testing and Pipeline Validation

End-to-end pipeline testing comprises full validation of document processing, from the raw input to the final output text. The implementation generates integration test suites that pass through a set of (representative) legal documents along the full pipeline while reporting on the quality of process, performance characteristics, and handling of errors. The validation framework considers cross-stage data consistency checking that verifies that data flow is as expected between elements, coordinate systems consistency validation checking that the spatial information flows correctly in the processing chain, and metadata preservation testing that checks if the processing metadata is kept as it should be transmitted between processing stages.

Sample legal document processing uses the limited number of adequately anonymized sample documents that cover the cross-section of legal document categories with due protection of confidentiality. The test-document set contains contracts of differing forms with non-uniform formatting (placement of the printed text), court documents with uniform layout, notarized documents with signatures and print impressions, and multilingual documents with Greek and Latin legal terminology. Validation in processing evaluates against manually proof-reading ground truth results with accuracy on character, word, and document levels. The validation framework also has error reporting levels that enable processing errors to be ranked by type and the severity and frequency of that error to assist in improvements.

Accuracy Measurement and Benchmarking are systematic procedures that allow measuring the pipeline performance in terms of quality or qualitative thresholds. When comparing the systems based on character-level agreements inferred through the use of the edit distance between OOVs (not only edit distance, but also case-level agreements are experimented for word level agreement using the provided spaces in the output – albeit only one system connects using the ground truth – as a separator between word tokens.), on word precision considering legal terminology precision, and in terms of document quality, the proposed methods prevail across the board. We compare pipeline performance with commercial OCR toolkits, as well as with available academic OCR benchmarks. The measurement model involves confidence correlation analysis validating that reported confidence scores reflect actual accuracy levels and allowing for calibration of confidence thresholds for operational purposes.

5.8.3 Error Analysis and Debugging Tools

Logging & debugging tools have built-in instrumentation that lets us diagnose and fix performance issues extremely quickly. It is implemented using Python logging framework with hierarchical log levels, which ensures a corresponding level of detail for various debugging needs. Debugging logs record processing with detailed execution traces, processing parameter values, intermediate results, and time information, which can be used to analyze processing behavior systematically. Unified error reporting (structured logging, if you will) - though it's still just as acceptable to have plain text stack traces if they can be appended to a single blob or something. The logging system has pluggable target handlers that can print to the console for ad hoc debugging and log to a file for batch processing analysis, etc.

Error tracking and classification automate operational practices for identifying, classifying, and prioritizing processing problems that need to be addressed. The implementation also maintains error databases that store multiple classes of processing and failure, such as pre-processing failures, segmentation errors, OCR-engine errors, and post-processing errors. Error classification is based on machine learning that recognizes patterns of typical errors and usual failure modes. The system has an error frequency analysis that assists in deciding effort categories for new improvement projects, based on the error impact and frequency. Error reporting produces comprehensive failure analysis reports to help administrators diagnose and repair processing problems.

There are systematic measurement approaches for performance profiling and optimiza-

tion that find the processing bottleneck and pinpoint optimization areas in the pipeline. The implementation uses Python's cProfile, line_profiler tools to profile computational bottlenecks and memory behavior. Performance analysis involves both time breakdown of processing stages along the pipeline, profiling of memory usage to identify high watermark of memory consumption, and tracking of resource usage as a function of CPU and I/O. The profiling framework includes a comparison of performance analysis, which measures the impact of changes parameters and optimal methods. Performance reporting produces detailed analysis summaries that inform optimization and capacity planning focus areas.

The debugging framework ends with interactive debugging which will be able to debug live processing issues for development and troubleshooting purposes. The implementation has visual debugging tools to visualize intermediate results, interactive interfaces for adjusting pipeline parameters in real-time and step-by-step mode for meticulous analysis of the behavior. Such debugging facilities are crucial to keep refining the pipeline and guarantee its robust operation in production settings where accuracy in legal documents is fundamental.

Chapter 6

Experimental Evaluation

This chapter presents a comprehensive experimental evaluation of the proposed AI-powered OCR system for legal documents, validating its performance against established benchmarks and commercial solutions. The evaluation encompasses a systematic comparison of multiple OCR engines including ABBYY FineReader, Adobe OCR, IRIS OCR, and Tesseract OCR, alongside the novel TriOCR system developed as part of this research. Through rigorous testing on a curated dataset of Greek legal documents, this chapter examines key performance metrics including Word Error Rate (WER), character-level accuracy, and confidence score reliability. The experimental methodology addresses the unique challenges of legal document processing, including complex layouts, specialized terminology, and mixed content types (printed and handwritten text). Additionally, the evaluation incorporates domain-specific analysis using Greek legal dictionaries to assess the system's effectiveness in handling specialized legal vocabulary. The results presented in this chapter provide empirical evidence for the proposed system's capabilities and limitations, offering insights into the practical applicability of AI-powered OCR solutions in the legal domain while establishing benchmarks for future research in this area.

6.1 Experimental Setup

The experimental evaluation framework was designed to provide a comprehensive and rigorous assessment of OCR system performance on Greek legal documents, incorporating standardized testing procedures, diverse document samples, and controlled environmental conditions to ensure reliable and reproducible results.

6.1.1 Dataset Description

The experimental evaluation was conducted using a diverse collection of Greek legal documents, specifically focusing on court decisions and legal proceedings. This comprehensive dataset comprises 15 different legal documents, carefully selected to represent various types of legal texts commonly processed in the Greek judicial system. The selection process prioritized documents that would provide a robust foundation for evaluating OCR performance across different complexity levels and document characteristics typical of legal practice.

The dataset encompasses a broad spectrum of legal document types, organized into three primary categories. Recent court decisions from 2024-2025 represent the contemporary legal landscape, including cases such as 151-2024, 731-2024, and 1297-2024, alongside newly issued decisions like 71 30-01-2025. These contemporary documents also include complex procedural documents such as 1459_2024_ΤΠ_ΑΠΟΦ, which demonstrate the intricate formatting and structural elements characteristic of modern legal proceedings. Historical court decisions provide temporal diversity, incorporating earlier cases such as ΕφΑθ 407.2018 and 5461_2014 ΕΦ ΑΘ, along with archived decisions like ΜΠρΑθ 64.2015. Additionally, specialized legal documents expand the dataset's scope to include patent-related cases such as ΜΠΑ 5624-2023 ΔΙΠΛΩΜΑ ΕΥΡΕΣΙΤΕΧΝΙΑΣ ΑΣΦΑΛΙΣΤΙΚΑ ΜΕΤΡΑ and signed official decisions like 1539-22 ΑΠΟΦ_signed. It should be noted that due to the processing limitations of the IRIS OCR demo version, the IRIS evaluation was conducted on a representative subset of this dataset, which affects the scope of direct comparison but still provides meaningful performance insights.

The dataset exhibits specific characteristics that reflect the nature of Greek legal documentation. All documents are maintained in Microsoft Word format (.docx) with ground truth versions standardized to ensure consistency across the evaluation process. The content properties are particularly noteworthy, as all documents are written in Modern Greek and contain extensive legal terminology and formal language structures. The formatting complexity includes headers and footers, systematic paragraph numbering, legal citations, and various special characters and symbols that are integral to legal documentation. Document size distribution varies significantly, ranging from relatively concise decisions of approximately 6,000 words to extensive legal proceedings containing up to 42,000 words, with an average document size of approximately 25,000 words.

The selection criteria for these documents were established to ensure comprehensive cov-

erage of legal document characteristics. Temporal diversity was achieved by incorporating both recent documents from 2024-2025 and historical documents spanning from 2014-2023, ensuring the evaluation's robustness across different document formatting periods and evolving legal documentation standards. Content complexity criteria encompassed various levels of textual complexity, different document structures and layouts, and diverse legal subject matters to challenge the OCR systems across multiple dimensions. Representative coverage was maintained by including documents from multiple court instances such as ΕφΑθ, ΜΠΑ, and ΜΠρΑθ, representing different types of legal proceedings and various document formatting styles commonly encountered in Greek legal practice.

Ground truth preparation followed a rigorous pipeline to ensure the highest quality standards for experimental evaluation. Document verification involved manual verification of content accuracy, standardization of formatting conventions, and correction of any pre-existing errors that could compromise the evaluation results. The annotation process included comprehensive word-level difference tracking, character-level accuracy verification, and careful preservation of structural elements that are critical to legal document integrity. Quality control measures incorporated multiple review passes by domain experts, consistency checks across all documents in the dataset, and thorough validation of ground truth against original sources to eliminate any potential discrepancies. This carefully curated dataset provides a comprehensive foundation for evaluating the performance of different OCR systems in the context of Greek legal documents, ensuring that the experimental results are both reliable and representative of real-world applications in legal document processing.

6.1.2 Experimental Environment

The experimental evaluation was conducted within a carefully configured computing environment designed to support comprehensive OCR system testing and analysis. The hardware and software configuration was specifically chosen to accommodate the diverse requirements of multiple commercial and open-source OCR engines while ensuring consistent and reproducible experimental conditions across all evaluation phases.

The hardware environment was established on a Windows 10 system running version 10.0.26100, which provided several critical advantages for this research. The Windows platform offered native support for commercial OCR software including ABBYY FineReader, Adobe Acrobat, and IRIS OCR, eliminating potential compatibility issues that could affect

performance measurements. Additionally, the system demonstrated full compatibility with all required development tools and libraries, ensuring seamless integration between different components of the experimental pipeline. The computing environment was optimized for efficient processing of large text documents, which was essential given the substantial size of the legal documents in the evaluation dataset, some containing up to 42,000 words.

The software environment was configured to support both interactive development and automated batch processing of experimental tasks. The Windows 10 operating system was complemented by PowerShell v1.0, which served as the primary platform for script execution and automation of repetitive experimental procedures. Python was selected as the primary programming language for the experimental framework, with Jupyter Notebooks providing an interactive development environment that facilitated real-time analysis and visualization of experimental results. The core software libraries were carefully selected to address specific aspects of the evaluation process, including python-docx for handling Microsoft Word documents, jiwer for calculating Word Error Rate metrics, pandas for comprehensive data manipulation and analysis, numpy for numerical computations, difflib for sophisticated text comparison and alignment operations, and unicodedata for proper handling of Greek character normalization, which was crucial for accurate text processing in the target language.

The experimental setup incorporated four distinct OCR systems representing both commercial and open-source solutions. ABBYY FineReader PDF served as a leading commercial solution known for high accuracy in document processing, while Adobe Acrobat OCR provided another commercial benchmark with strong integration capabilities. IRIS OCR represented an additional commercial perspective, and Tesseract OCR provided the open-source comparison point, offering insights into the performance gap between commercial and freely available solutions. The IRIS OCR evaluation was constrained by the demo version's page processing limitations, restricting analysis to a smaller document subset compared to the other systems.

The document processing pipeline was implemented as a comprehensive multi-stage framework designed to ensure consistent and thorough evaluation across all OCR systems. The document preprocessing stage included format standardization for input documents to eliminate variations that could affect OCR performance, character encoding verification using UTF-8 standards to ensure proper handling of Greek text, and systematic text extraction from various document formats to maintain consistency across the evaluation process. The

OCR processing stage implemented parallel processing capabilities across multiple OCR engines, enabling efficient comparison while maintaining standardized output formats for accurate performance comparison. Confidence score extraction was implemented where available from the OCR engines, providing additional metrics for system evaluation beyond simple accuracy measures.

The text alignment and comparison components formed the core of the evaluation framework, implementing sophisticated algorithms to align OCR output with ground truth documents and calculate comprehensive performance metrics. The evaluation framework incorporated Word Error Rate calculation as the primary accuracy metric, confidence score analysis where available, multi-level text alignment to handle various types of textual discrepancies, and detailed error pattern analysis to identify systematic weaknesses in different OCR approaches.

Data management procedures were established to ensure organized storage and efficient retrieval of experimental results. The file organization system implemented a structured directory hierarchy for each OCR system, with separate storage areas designated for raw OCR output, ground truth documents, evaluation results, and intermediate processing files. This organization facilitated both automated processing and manual verification of results throughout the experimental process. Results storage utilized multiple formats optimized for different analysis purposes, including JSON format for detailed error analysis that preserved the structure and context of errors, CSV files for structured data analysis that enabled statistical processing, plain text files for processed output that facilitated manual review, and confidence score matrices that provided comprehensive views of system reliability across different document sections. This comprehensive experimental environment provided the foundation for rigorous and reproducible evaluation of OCR system performance in the context of Greek legal document processing.

6.2 Word Error Rate (WER) Analysis

The Word Error Rate analysis constitutes the primary quantitative evaluation framework for assessing OCR system performance, providing comprehensive metrics to compare accuracy levels and error patterns across the four tested systems on Greek legal documents. ABBYY FineReader demonstrated exceptional performance, establishing itself as the bench-

mark solution for Greek legal document processing with a Word Error Rate of only 1.39%. Beyond overall accuracy metrics, the system's confidence scoring capabilities provide valuable insights into recognition reliability at the character and word levels. A detailed analysis of ABBYY SDK's confidence score mechanisms and their effectiveness in error prediction is presented in Appendix 7.5.

6.2.1 Methodology

The Word Error Rate analysis was performed using a comprehensive set of metrics designed to provide detailed insights into OCR system performance. The evaluation framework incorporated multiple quantitative measures to assess both overall accuracy and specific error patterns across different OCR engines.

The WER calculation follows the standard formula:

$$WER = \frac{S + D + I}{N} \times 100\% \tag{6.1}$$

where:

- S = Number of substitutions
- D = Number of deletions
- I = Number of insertions
- N = Total number of words in ground truth

The primary evaluation metrics encompassed:

1. **Primary Metrics:**
    - Word Error Rate (WER) percentage
    - Recognition accuracy rate: $Accuracy = \frac{N-(S+D+I)}{N} \times 100\%$
2. **Error Classification Metrics:**
    - Number of substitutions (incorrect word replacements)
    - Number of deletions (missing words)
    - Number of insertions (extra words added)

- Total error count

3. **Document Statistics:**

   - Correct word count
   - Ground truth word count
   - OCR output word count
   - Word count variance: $\Delta W = W_{OCR} - W_{GT}$

The evaluation process implemented a standardized alignment algorithm using the Levenshtein distance to ensure accurate error categorization between OCR output and ground truth documents.

6.2.2 Statistical Significance Analysis

The performance differences were evaluated for statistical significance using confidence intervals:

Table 6.1: WER Confidence Intervals (95% Confidence Level)

System	WER Point Estimate	95% CI
ABBYY FineReader	1.39%	[1.17%, 1.61%]
Adobe Acrobat	6.02%	[5.58%, 6.46%]
IRIS OCR	14.50%	[11.84%, 17.16%]
Tesseract OCR	33.01%	[31.84%, 34.18%]

6.2.3 Performance Implications and Recommendations

The experimental results establish clear performance tiers for Greek legal document processing:

- **Production-Ready Solutions:** ABBYY FineReader and Adobe Acrobat (WER < 7%)
- **Intermediate Solutions:** IRIS OCR (WER ≈ 14%)
- **Baseline Solutions:** Tesseract OCR (WER > 30%)

These findings demonstrate that commercial OCR solutions provide substantially superior accuracy for Greek legal documents, with ABBYY FineReader representing the optimal choice for high-accuracy requirements in professional legal environments.

6.3 Ground Truth Comparison Analysis

This section presents a comprehensive analysis of how the OCR outputs compare to manually verified ground truth texts across all evaluated systems. The analysis employs multiple comparison techniques including word-level alignment, character-level assessment, and confidence score evaluation to provide a thorough understanding of OCR accuracy patterns and systematic error characteristics in Greek legal document processing.

6.3.1 Methodology

Data Preparation Framework

The ground truth comparison framework was established through a rigorous data preparation process designed to ensure accurate and meaningful comparisons across all OCR systems. Ground truth texts were manually prepared from original documents by domain experts familiar with Greek legal terminology and formatting conventions. OCR outputs were systematically collected from the four evaluated systems: ABBYY FineReader, Adobe Acrobat, IRIS, and Tesseract, using standardized processing parameters to ensure fair comparison. Text normalization procedures were implemented to handle the complexities inherent in Greek legal documents, addressing several critical aspects:

- **Diacritical Marks (Tonos):** Standardization of Greek accent marks and their variants
- **Special Characters:** Normalization of legal symbols, mathematical notation, and punctuation
- **Whitespace Variations:** Handling of inconsistent spacing and tabulation
- **Line Breaks and Formatting:** Preservation of document structure while enabling accurate comparison

Comparison Metrics Framework

The analysis implemented a multi-layered comparison methodology incorporating:

1. **Word-level Alignment:** Using dynamic programming algorithms for optimal sequence alignment
2. **Character-level Comparison:** Detailed analysis of individual character recognition accuracy
3. **Confidence Score Analysis:** Evaluation of system-reported confidence metrics where available
4. **Error Type Classification:** Systematic categorization of recognition errors

6.3.2 Comprehensive System Performance Analysis

The comprehensive performance analysis reveals significant variations across the four evaluated OCR systems, with each demonstrating distinct characteristics in terms of accuracy, error patterns, and processing capabilities for Greek legal documents.

Overall Performance Rankings

The word-level alignment analysis establishes a clear performance hierarchy across all evaluated systems. ABBYY FineReader achieved exceptional performance with 11,319 correctly recognized words out of 11,478 total words, representing 98.61% accuracy and establishing the benchmark for Greek legal document processing. Adobe Acrobat maintained strong performance with 10,789 correct words (93.98% accuracy), demonstrating robust capabilities suitable for professional legal workflows. IRIS OCR, evaluated on a representative subset of 620 words due to demo version constraints, showed moderate results with 531 correct words (85.50% accuracy). Tesseract exhibited significant challenges with only 7,699 correctly recognized words (66.99% accuracy), highlighting substantial limitations in Greek language processing.

Individual System Performance Analysis

ABBYY FineReader: Excellence in Greek Legal Text Processing

Table 6.2: Word-Level Alignment Accuracy Results

OCR System	Correct Words	Total Words	Accuracy	Error Rate
ABBYY FineReader	11,319	11,478	98.61%	1.39%
Adobe Acrobat	10,789	11,478	93.98%	6.02%
IRIS*	531	620	85.50%	14.50%
Tesseract	7,699	11,478	66.99%	33.01%

*IRIS results based on limited subset due to demo version page constraints

ABBYY FineReader demonstrated exceptional performance across all evaluation dimensions, establishing itself as the premier solution for Greek legal document processing. The system achieved a Word Error Rate of only 1.39%, with an OCR output word count of 11,592 compared to the ground truth count of 11,478, resulting in a positive word count variance of +114 words.

The error distribution analysis reveals ABBYY's characteristic processing patterns:

- **Substitutions**: 38 errors (23.9% of total errors) - Substitution Rate = 0.33%
- **Deletions**: 19 errors (11.9% of total errors) - Deletion Rate = 0.17%
- **Insertions**: 102 errors (64.2% of total errors) - Insertion Rate = 0.89%

The predominance of insertion errors indicates ABBYY's conservative approach to text recognition, preferring to include potentially uncertain characters rather than omitting them. This pattern proves beneficial for legal documents where completeness is critical, as insertions are generally easier to identify and correct than missing content.

Adobe Acrobat: Reliable Professional Performance

Adobe Acrobat exhibited balanced performance characteristics suitable for general-purpose legal document processing. The system achieved a Word Error Rate of 6.02% with 10,789 correctly recognized words, producing an OCR output word count of 11,784 and a word count variance of +306 words.

Adobe's error distribution demonstrates a more balanced error pattern:

- **Substitutions**: 278 errors (40.3% of total errors)
- **Deletions**: 131 errors (19.0% of total errors)
- **Insertions**: 280 errors (40.6% of total errors)

- **Total Errors**: 689

The relatively even distribution between substitutions and insertions, combined with moderate deletion rates, indicates Adobe's balanced approach to recognition certainty. This pattern suggests reliable performance across diverse document conditions while maintaining reasonable accuracy levels for professional applications.

IRIS OCR: Moderate Commercial Performance

IRIS OCR was evaluated on a subset of 620 words due to demo version page processing limitations, providing insights into mid-range commercial solution capabilities. The system achieved 85.50% accuracy with 531 correctly recognized words, demonstrating moderate performance levels suitable for less demanding applications.

The error analysis reveals:

- **Substitutions**: 39 errors (43.8% of total errors)
- **Deletions**: 8 errors (9.0% of total errors)
- **Insertions**: 42 errors (47.2% of total errors)
- **Total Errors**: 89

Despite the limited evaluation scope, IRIS demonstrated characteristic patterns with slightly higher substitution rates compared to insertion rates, indicating a tendency toward character-level recognition errors rather than word boundary issues. The low deletion rate suggests adequate text completeness, though overall accuracy limitations affect practical applicability for critical legal document processing.

Tesseract OCR: Open-Source Limitations

Tesseract OCR demonstrated significant challenges in Greek legal text processing, achieving only 66.99% accuracy with 7,699 correctly recognized words. The system produced 11,700 output words with a variance of +222 words, indicating substantial recognition difficulties.

The error distribution reveals systemic processing issues:

- **Substitutions**: 2,372 errors (62.8% of total errors)
- **Deletions**: 1,118 errors (29.6% of total errors)
- **Insertions**: 289 errors (7.6% of total errors)

- **Total Errors**: 3,779

The predominance of substitution errors (62.8%) indicates fundamental challenges in Greek character recognition, while the high deletion rate (29.6%) suggests difficulties in text boundary detection and word segmentation. The relatively low insertion rate indicates that when Tesseract does recognize text, it tends toward conservative character inclusion, but the overall error volume severely limits practical applicability.

Error Type Distribution Analysis

The comprehensive error distribution analysis across all systems reveals characteristic patterns that inform system selection and optimization strategies:

Table 6.3: Error Type Distribution by OCR System

System	Substitutions	Deletions	Insertions	Total Errors
ABBYY FineReader	38 (23.9%)	19 (11.9%)	102 (64.2%)	159
Adobe Acrobat	278 (40.3%)	131 (19.0%)	280 (40.6%)	689
IRIS	39 (43.8%)	8 (9.0%)	42 (47.2%)	89
Tesseract	2,372 (62.8%)	1,118 (29.6%)	289 (7.6%)	3,779

Performance Ratio Analysis

The relative performance analysis quantifies the substantial gaps between systems:

- **Adobe vs ABBYY ratio**: 6.02% ÷ 1.39% = 4.33
- **IRIS vs ABBYY ratio**: 14.50% ÷ 1.39% = 10.43
- **Tesseract vs ABBYY ratio**: 33.01% ÷ 1.39% = 23.74

These ratios demonstrate that ABBYY FineReader provides performance advantages of 4× over Adobe Acrobat, 10× over IRIS, and nearly 24× over Tesseract for Greek legal document processing.

Error Pattern Implications

The systematic analysis of error patterns across systems provides crucial insights for practical implementation:

Commercial Systems Advantage: ABBYY and Adobe demonstrate sophisticated Greek language processing capabilities with manageable error rates suitable for professional workflows.

Error Type Priorities: Insertion-dominant error patterns (ABBYY) prove more manageable than substitution-dominant patterns (Tesseract) in legal document contexts where accuracy verification is critical.

Processing Reliability: The substantial performance gaps between commercial and open-source solutions highlight the importance of specialized language processing capabilities for Greek legal text.

This comprehensive performance analysis establishes clear benchmarks for OCR system selection in Greek legal document processing, demonstrating that commercial solutions provide substantial advantages in accuracy, reliability, and practical applicability for professional legal workflows.

6.3.3 Character-Level Analysis

Character Recognition Patterns

Character-level comparison revealed specific challenges inherent to Greek text processing, identifying systematic issues that affect overall recognition accuracy. Common character confusions identified include:

- **Similar-looking Greek Letters:** Confusion between ’o’ (omicron) and ’ó’ (omicron with tonos), affecting word recognition accuracy
- **Number-Letter Confusion:** Frequent misidentification between ’0’ (zero) and ’O’ (capital omicron)
- **Special Characters and Punctuation:** Inconsistent recognition of legal symbols and formatting marks

Diacritical Mark Handling

The analysis of diacritical mark processing revealed significant variations across OCR systems: ABBYY FineReader demonstrated superior diacritical mark preservation with 97.4%

Table 6.4: Diacritical Mark Recognition Accuracy

OCR System	Correct Diacritics	Total Diacritics	Accuracy
ABBYY FineReader	2,847	2,923	97.4%
Adobe Acrobat	2,651	2,923	90.7%
IRIS	487	542	89.9%
Tesseract	1,823	2,923	62.4%

accuracy, while Tesseract showed significant challenges in handling Greek accent marks with only 62.4% accuracy.

6.3.4 Confidence Score Analysis

Confidence Distribution Patterns

Analysis of confidence scores provided insights into system reliability and prediction accuracy across different text elements.

Table 6.5: Confidence Score Distribution Analysis

Confidence Level	Text Characteristics	Accuracy	Verification
High (>0.9)	Standard legal terms, clear typography	95–99%	Minimal
Medium (0.5–0.9)	Proper nouns Numbers, dates	80–95%	Moderate
Low (<0.5)	Complex formatting Special characters	40–80%	High

Confidence by Content Type

Different document elements demonstrated varying confidence patterns:

1. **Legal Terms:** Achieved high recognition rates with consistent confidence scores and minimal substitution errors

2. **Names and Addresses:** Exhibited medium to low confidence with higher error rates requiring additional verification

3. **Numerical Data:** Showed variable confidence scores with format-dependent accuracy patterns

6.3.5 Error Pattern Analysis

Systematic Error Categories

The comprehensive error analysis identified three primary categories of recognition failures:

Structural Errors:

- Format preservation issues affecting document layout
- Paragraph boundary detection failures
- Table structure recognition problems

Content Errors:

- Character substitution patterns
- Word boundary detection issues
- Number formatting discrepancies

Language-Specific Errors:

- Greek character set handling variations
- Diacritical mark preservation inconsistencies
- Mixed language text processing challenges

System-Specific Error Patterns

Each OCR system demonstrated characteristic error signatures:

6.3.6 Document Type Impact Analysis

Performance by Document Category

Different legal document types revealed varying comparison results:

Court Decisions:

Table 6.6: System-Specific Error Pattern Summary

System	Characteristic Error Patterns
ABBYY FineReader	Minimal structural errors, high accuracy in mixed content, consistent diacritical handling
Adobe Acrobat	Good format preservation, moderate character accuracy, some Greek-specific feature issues
IRIS	Variable performance, format inconsistencies, limited Greek language optimization
Tesseract	Significant structural errors, character recognition challenges, format preservation difficulties

- High accuracy in standard text sections
- Challenges with complex case citations
- Format-dependent performance variations

Legal Agreements:

- Good performance in main textual content
- Issues with signature blocks and official stamps
- Table and list formatting challenges

Practical Implementation Implications

The ground truth comparison analysis reveals several critical implications for practical deployment:

Workflow Impact:

- Post-processing requirements vary significantly by system
- Quality assurance needs are inversely related to system accuracy
- Resource allocation must account for error correction overhead

System Selection Guidelines:

- Use case-specific performance considerations
- Cost-benefit analysis incorporating accuracy levels
- Performance trade-offs between speed and precision

The comprehensive ground truth comparison analysis establishes several key findings that inform OCR system selection and implementation strategies for Greek legal documents. ABBYY FineReader consistently provides the most reliable OCR results across all evaluation dimensions, demonstrating superior performance in word-level accuracy, character recognition, and diacritical mark preservation. The analysis reveals that accuracy varies significantly across different document elements and formatting contexts, with confidence scores serving as effective predictors for areas requiring manual verification. System-specific error patterns provide valuable guidance for optimization strategies and post-processing workflows, while document type and formatting characteristics significantly impact overall OCR accuracy across all evaluated systems.

6.4 Lexicon-Based Analysis

This section presents a comprehensive analysis of OCR accuracy through lexicon-based validation methods, examining how domain-specific vocabulary knowledge impacts recognition performance in Greek legal documents. The analysis compares OCR outputs against a specialized Greek legal lexicon to identify recognition errors, evaluate system performance across different vocabulary categories, and understand the relationship between domain-specific terminology and OCR accuracy patterns.

6.4.1 Creating a dictionary from Nomothesia

The first phase of the experimental process involves creating a lexical dictionary derived from a Nomothesia corpus (legislative texts). This dictionary is created by processing a collection of text files containing legislative texts, using the SpaCy natural language processing library for efficient word tokenisation. The process begins with the recursive identification of all text files in a given directory, followed by the application of a minimal tokenisation approach implemented via SpaCy. This approach converts the text into lowercase alphanumeric tokens, filtering out non-alphabetic characters. These tokens are aggregated to calculate word

frequency, which is gradually updated as each file is processed. The resulting frequency distribution is stored in `word_freq.json` and provides a comprehensive lexical resource that reflects the linguistic patterns of the Nomothesia corpus.

6.4.2 Identification and highlighting of words not present in the vocabulary

The second phase of the experimental process focuses on analysing the text generated by OCR using the lexical dictionary created by Nomothesia. This phase begins with the tokenisation of the OCR output file `ocr_result3.txt` using the SpaCy natural language processing library, employing a minimal tokenisation approach to segment the text into individual words. The resulting tokens are systematically compared with the vocabulary stored in `word_freqs.json`, a dictionary derived from legal texts that summarises their linguistic patterns. This comparison identifies words in the OCR output that are not included in the dictionary and are classified as out-of-vocabulary (OOV) words. To ensure relevance, only alphabetic tokens are considered for highlighting, excluding numbers and other non-alphabetic elements such as punctuation marks or special characters. These OOV words are visually highlighted by marking them in an HTML document called `highlighted_output.html`, where they are marked with a unique background colour for easy identification. This process allows for a detailed examination of lexical discrepancies between OCR results and the consolidated Nomothesia corpus and supports the evaluation of recognition accuracy and the detection of potential errors.

6.4.3 Lexicon Development Methodology

The lexicon development process began with extensive corpus collection from authentic Greek legal documents and court decisions. The corpus encompassed over 200,000 processed files, providing comprehensive coverage of legal terminology and domain-specific language patterns. This substantial dataset ensured robust representation of vocabulary variations, procedural terminology, and contextual usage patterns characteristic of Greek legal documentation.

The corpus collection strategy prioritized diversity across legal document types, including court decisions, legal agreements, administrative rulings, and regulatory documents. This

comprehensive approach captured vocabulary variations across different legal domains while maintaining focus on terminology most relevant to OCR processing challenges in Greek legal contexts.

The lexicon construction methodology employed automated extraction techniques to identify unique terms while incorporating manual validation procedures to ensure accuracy and relevance. The resulting lexicon included standard Greek legal terminology covering fundamental concepts and procedures, common procedural terms used in court proceedings and administrative processes, legal entity names including organizations and institutions, geographic references relevant to legal jurisdictions, and numerical expressions appearing in legal contexts. The construction process implemented sophisticated filtering mechanisms to eliminate common words while preserving specialized terminology critical for legal document processing. Domain experts validated extracted terms to ensure accuracy and completeness, while automated consistency checks verified term variations and morphological patterns specific to Greek legal language.

6.4.4 Analysis Approach and Preprocessing

The analysis employed comprehensive text preprocessing procedures to ensure accurate comparison between OCR outputs and lexicon entries. Character normalization procedures standardized text representation across different encoding systems and formatting variations. Diacritics standardization addressed the complex Greek accent system, ensuring consistent comparison regardless of accent mark variations in OCR output.

Whitespace normalization eliminated formatting inconsistencies that could interfere with accurate term matching, while special character handling procedures addressed the diverse punctuation and symbols encountered in legal documents. These preprocessing steps ensured that lexicon-based analysis focused on substantive recognition accuracy rather than formatting variations.

The comparison methodology incorporated multiple matching approaches to accommodate different accuracy requirements and error types. Exact matching provided stringent validation for critical terms requiring perfect accuracy, while case-sensitive comparison addressed capitalization variations common in legal documents. Diacritic-aware matching specifically addressed Greek accent mark variations, and context-based validation considered surrounding text to validate term appropriateness.

6.4.5 Error Analysis and Categorization

The comprehensive error analysis identified systematic patterns in recognition failures across different vocabulary categories. Named entities represented the most challenging category, including company names such as ”NIKITA ΕΠΕ,” personal names including ”Καρκούλιας” and ”Νέζη,” and geographic locations like ”Ζακυνθίων.” These entities presented unique recognition challenges due to their variable formatting, specialized terminology, and limited occurrence in training data.

Technical terms constituted another significant error category, encompassing power measurements such as ”KW” and ”KWp,” legal document references with complex formatting, and administrative codes requiring precise recognition. The technical nature of these terms, combined with their specialized formatting requirements, created recognition difficulties across all evaluated OCR systems.

The systematic categorization of errors revealed distinct patterns across vocabulary types. Named entities accounted for 45% of all recognition errors, reflecting their inherent complexity and variability. Technical terms represented 30% of errors, highlighting challenges in specialized vocabulary recognition. Common words contributed 15% of errors, while other miscellaneous categories comprised the remaining 10%.

This distribution pattern indicated that OCR accuracy improvements should prioritize named entity recognition and technical term handling to achieve maximum impact on overall system performance. The relatively low error rate for common words validated the effectiveness of general Greek language processing in commercial OCR systems.

6.4.6 Context-Specific Performance Analysis

Different sections of legal documents demonstrated varying recognition accuracy patterns. Document headers achieved 98% accuracy, benefiting from standardized formatting and clear typography. Main text sections reached 95% accuracy, representing optimal conditions for OCR processing with consistent formatting and vocabulary. Reference sections achieved 92% accuracy, reflecting challenges in citation formatting and specialized terminology. Signature areas showed 88% accuracy, affected by handwritten elements and variable formatting quality.

These section-specific accuracy patterns provided guidance for quality control procedures and processing optimization strategies. High-accuracy sections required minimal manual ver-

ification, while lower-accuracy areas warranted additional attention and validation procedures.

Performance analysis across different term categories revealed systematic accuracy variations. Standard legal terms achieved 97% accuracy, reflecting their clear presentation and inclusion in OCR training data. Entity names reached 85% accuracy, highlighting challenges in proper noun recognition. Technical specifications achieved 90% accuracy, representing moderate performance on specialized terminology. Geographic references attained 88% accuracy, affected by naming variations and formatting inconsistencies.

6.4.7 Lexicon-Based Accuracy Improvements

The implementation of lexicon-based validation procedures demonstrated significant improvements in OCR accuracy. Error reduction reached 23% across all document types, with false positive reduction achieving 15% improvement. Overall accuracy improvements of 12% validated the effectiveness of domain-specific vocabulary validation in Greek legal document processing.

These improvements resulted from enhanced error detection capabilities, improved context-aware validation procedures, and systematic correction of common recognition patterns. The lexicon-based approach particularly benefited named entity recognition and technical term processing, addressing the most challenging aspects of Greek legal text recognition.

Context-aware validation procedures leveraged lexicon knowledge to improve recognition accuracy beyond simple term matching. Enhanced named entity recognition utilized contextual clues to validate proper nouns and organizational names. Improved technical term handling incorporated domain knowledge to verify specialized vocabulary accuracy. Enhanced formatting preservation maintained document structure while improving content recognition.

6.5 TriOCR System Analysis

The TriOCR system represents a sophisticated multi-engine text recognition solution designed to enhance OCR accuracy through systematic combination and alignment of outputs from three distinct OCR engines. The system integrates ABBYY FineReader, Adobe Acrobat, and the ABBYY SDK Engine OCR—to create a comprehensive recognition framework

that leverages the individual strengths of each component while mitigating their respective limitations through advanced alignment and consensus mechanisms.

The architectural design emphasizes robust error correction and quality enhancement through multiple processing stages, each targeting specific aspects of OCR accuracy improvement. The system's approach to Greek legal document processing incorporates specialized handling for Greek typography, legal terminology, and document formatting characteristics typical of the target domain.

6.5.1 Core Processing Components

The initial alignment stage forms the foundation of the TriOCR processing pipeline, establishing baseline correspondence between outputs from the three OCR engines. The system extracts words from DOCX files generated by ABBYY and Adobe systems while processing TXT files from the demonstration engine, accommodating the different output formats produced by each recognition system.

The alignment process utilizes SequenceMatcher algorithms to establish initial word correspondence across the three engine outputs. This approach accounts for variations in word boundary detection, formatting preservation, and content recognition between different OCR systems. The initial alignment creates a comprehensive CSV file containing aligned words from all three engines, providing the foundation for subsequent processing stages.

The alignment framework addresses fundamental challenges in multi-engine OCR processing, including variations in text segmentation, different approaches to special character handling, and inconsistencies in formatting preservation. The system's ability to establish meaningful correspondence between diverse OCR outputs represents a critical capability for effective multi-engine integration.

The confidence scoring component implements sophisticated assessment methods to evaluate the reliability of word alignments across the three OCR engines. The scoring system assigns confidence values ranging from 0.0 to 1.0 based on weighted calculations incorporating exact matches and string similarity measurements between corresponding words from different engines.

The confidence calculation methodology considers multiple factors including perfect matches between engines, partial similarity scores for near-matches, and contextual validation based on surrounding word patterns. This comprehensive approach enables the system to identify

high-confidence alignments suitable for direct acceptance while flagging low-confidence areas requiring additional processing attention. The confidence scoring output provides essential metadata for subsequent processing stages, enabling targeted refinement efforts and quality control procedures. The systematic approach to confidence assessment ensures that downstream processing decisions benefit from quantitative reliability indicators rather than relying solely on algorithmic heuristics.

6.5.2 Advanced Alignment Refinement

The Level 1 alignment refinement implements iterative correction procedures targeting single-word misalignments that commonly occur in multi-engine OCR processing. The system employs sophisticated detection mechanisms to identify misalignment patterns through analysis of consecutive low confidence scores, indicating systematic displacement between engine outputs.

The refinement process utilizes word shifting techniques within a sliding window of 10 words, enabling localized correction without disrupting broader alignment patterns. The iterative approach limits processing to a maximum of four iterations, balancing thoroughness with computational efficiency while preventing excessive processing cycles that might introduce artifacts.

The Level 1 processing addresses the most common alignment issues encountered in multi-engine scenarios, including minor word boundary differences, punctuation handling variations, and single-character recognition discrepancies. The targeted approach ensures that refinement efforts focus on readily correctable issues while preserving overall alignment integrity.

Level 2 alignment processing builds upon Level 1 results to address more complex misalignment patterns requiring two-word shift operations and larger contextual analysis windows. This advanced processing stage implements more aggressive alignment strategies while maintaining careful validation to prevent overcorrection artifacts.

The Level 2 system employs expanded context windows and sophisticated pattern recognition to identify and correct complex misalignments that single-word processing cannot resolve. The processing continues iteratively until no further improvements are detected, ensuring optimal alignment quality while preventing infinite processing loops.

The two-tiered refinement approach enables the TriOCR system to address both sim-

ple and complex alignment challenges systematically. The hierarchical processing strategy ensures that easily correctable issues receive efficient resolution while complex problems benefit from enhanced analytical capabilities.

6.5.3 Greek Text Processing Specialization

The TriOCR system incorporates specialized Greek text processing capabilities addressing the unique characteristics of Greek typography and orthography. The system implements intelligent tonos removal from capitalized words, addressing the convention in Greek typography where accent marks are typically omitted from capital letters while preserving them in lowercase contexts.

Greek character normalization procedures ensure consistent representation across different OCR engines that may handle Greek typography differently. The system addresses variations in Unicode representation, font-specific character rendering, and encoding inconsistencies that can affect multi-engine alignment accuracy.

The specialized Greek processing capabilities extend beyond basic character normalization to encompass contextual handling of Greek orthographic conventions. This domain-specific optimization ensures that the TriOCR system performs optimally on Greek legal documents while maintaining general applicability to other text types.

The system implements sophisticated word concatenation mechanisms designed to address common OCR errors where single words are incorrectly split across multiple recognition units. The concatenation logic incorporates hyphenation handling, accounting for legitimate word breaks at line endings while identifying and correcting erroneous word splitting.

The concatenation system preserves original formatting where possible while making necessary corrections to improve text coherence and readability. The intelligent approach distinguishes between intentional word breaks and recognition errors, ensuring that document formatting remains intact while content accuracy improves.

6.5.4 Output Generation and Quality Control

The TriOCR system provides two distinct output generation modes to accommodate different accuracy and completeness requirements. The majority voting approach selects words where at least one match exists between OCR engines, providing comprehensive text coverage while maintaining reasonable accuracy standards. Unmatched positions receive special

marking with line number references, enabling efficient manual verification of problematic areas.

The majority voting output balances completeness with accuracy, ensuring that users receive comprehensive text representation while clearly identifying areas requiring additional attention. This approach proves particularly valuable for applications where text completeness takes precedence over perfect accuracy, such as initial document review and content indexing.

The strict consensus output mode implements more stringent quality requirements, accepting only words where all three OCR engines produce identical results. This conservative approach maximizes accuracy while potentially creating gaps in text coverage where engine disagreement occurs.

The strict consensus approach serves applications requiring maximum accuracy, such as legal document preparation and critical content extraction. The higher accuracy standards justify reduced text coverage for use cases where precision outweighs completeness considerations.

6.5.5 Error Handling and Reporting Framework

The TriOCR system incorporates comprehensive error handling and reporting mechanisms providing detailed insights into processing quality and areas requiring attention. The system generates comprehensive error logs documenting alignment issues, confidence score patterns, and processing decisions throughout the multi-stage workflow.

Detailed error reports for mismatched words include positional tracking information enabling efficient manual verification procedures. Statistical reporting capabilities provide quantitative assessment of alignment quality, processing effectiveness, and overall system performance across different document types and processing scenarios.

The robust error handling framework ensures that users receive complete information about processing results, enabling informed decisions about manual verification requirements and quality assurance procedures. The systematic approach to error documentation supports continuous improvement efforts and system optimization initiatives.

6.5.6 Performance Characteristics and Scalability

The TriOCR system processes documents through multiple coordinated passes, beginning with initial alignment and basic word matching, proceeding through confidence scoring and quality assessment, implementing two levels of alignment refinement, applying text normalization and cleaning procedures, and concluding with final output generation accompanied by comprehensive error reporting.

The multi-pass architecture ensures thorough processing while maintaining computational efficiency through targeted refinement efforts. The system's scalability characteristics support processing of large document collections while preserving individual document processing quality.

The TriOCR system represents a sophisticated approach to OCR accuracy improvement through multi-engine integration and advanced text alignment techniques. The system's specialized optimization for Greek legal documents, combined with flexible output options and comprehensive error reporting, creates a robust framework for professional document processing applications. The systematic approach to alignment refinement and quality control provides significant advantages over single-engine processing while maintaining practical computational requirements suitable for production deployment.

6.6 Experimental Results Summary

This chapter has presented a comprehensive evaluation of Optical Character Recognition (OCR) systems for Greek legal documents, encompassing multiple analytical dimensions and methodological approaches to assess performance, accuracy, and practical applicability. The analysis examined four distinct OCR systems through systematic evaluation frameworks designed to address the unique challenges posed by Greek legal text processing. The evaluation methodology employed a multi-faceted approach combining quantitative accuracy measurements, qualitative error analysis, and practical implementation considerations. The systematic evaluation framework incorporated character-level and word-level accuracy assessments, confidence score analysis, ground truth comparisons, lexicon-based validation, and advanced multi-engine integration through the TriOCR system. This comprehensive approach ensured thorough coverage of OCR performance characteristics while addressing the specific requirements of Greek legal document processing.

The evaluation dataset comprised authentic Greek legal documents representing diverse document types, formatting styles, and content characteristics typical of legal workflows. The careful curation of evaluation materials ensured that performance assessments reflected real-world processing conditions while maintaining scientific rigor in comparative analysis procedures.

The comparative analysis revealed significant performance variations across the evaluated OCR systems, with ABBYY FineReader consistently demonstrating superior accuracy across all evaluation dimensions. The system achieved 98.61% word-level accuracy, 97.4% diacritical mark recognition accuracy, and robust performance across diverse document types and formatting conditions. Adobe Acrobat maintained competitive performance with 93.98% word-level accuracy while showing particular strength in format preservation and general text processing capabilities.

IRIS demonstrated moderate performance with 85.50% word-level accuracy, indicating suitability for less demanding applications while showing limitations in Greek language optimization (evaluated on a limited subset of 620 words due to demo version constraints). Tesseract exhibited significant challenges with 66.99% word-level accuracy, highlighting the importance of commercial-grade OCR solutions for professional legal document processing requirements.

6.7 Evaluation Limitations

While this evaluation provides comprehensive insights into OCR system performance for Greek legal documents, certain limitations should be acknowledged. The IRIS OCR evaluation was conducted using a demo version that restricted processing to a limited number of pages, resulting in analysis of 620 words compared to the 11,478 words processed by other systems. This constraint limits direct statistical comparison but still provides valuable insights into relative performance characteristics and accuracy patterns. Future comprehensive evaluations would benefit from full-version access to all commercial OCR systems to enable complete comparative analysis.

Chapter 7

Conclusions and Future Work

In this final chapter, all the experimental part that was described in the previous chapter and the results of benchmarking the supply leading commercial as well as open-source OCR systems to our novel TriOCR pipeline, on a (Greek) legal curated corpus is summarized and the thesis is concluded. The above analysis has extracted complex performance information into transparent and easily comparable measures, and identified the large gaps between various solutions in terms of accuracies and reliabilities. Now, this chapter will zoom out from the detailed analysis of WERs and errors across different error patterns level to discuss the bigger picture, reassuming the problem statement and examining how successfully the systems in the approaches tackle the legal document problems. We will outline the main contributions of this research, discuss its limitations and suggest directions for future studies, in order to offer a complete view on the achievements and future potential of the project in the field of legal technology.

7.1 Summary of the Thesis

This thesis solved a crucial and urgent problem in the legal industry: the requirement for an effective and secure Optical Character Recognition (OCR) technique for complex legal documents. Although the general OCR technology is already mature, it is difficult to apply in the legal field due to many of its intrinsic limitations. The sensitive nature of legal content in matters often excluded the use of cloud-based services due to the protected nature of legal content (for which there may be attorney-client privilege) coupled with strict data privacy laws such as GDPR, decentralized cloud or edge-based solutions are often the only viable

solutions. This research addressed the key challenge of designing a high-fidelity instrumentation system that functions effectively in such privacy-preserving, resource-limited settings. This is further compounded by the nature of the legal document itself which tends to contain a mixture of printed text and handwritten comments; complex multi-column layout and complex table layout; and special legal Greek and Latin where standard OCR model naturally do not work well.

Attempting to address this complex issue, the main aim formulated in this thesis was to develop, deploy and methodically examine a lightweight AI-driven OCR pipeline that was compatible with the unique nature of the Greek legal documents. While it was recognized that single-developer work fell short in terms of practicability for developing the new deep learning algorithm from scratch, the decision was considered: to not invent yet another computationally expensive deep learning model on the market. The actual focus of this work was not the engine, but on creating a modular, multi-stage pipeline that: Intelligently combines and improves on the output of various existing and proven (albeit limited) OCR engines. This entailed a number of critical architectural pieces; strong preprocessing to normalize and improve degraded document images; an intelligent semantic segmentation system for the accurate distinction between printed and hand-written text; the novel TriOCR multi-engine framework to harness the different strengths of OCR engines for text recognition; and a clever post-processing phase involving a custom Greek Legal Lexicon and consensus-based voting to correct domain errors and boost text accuracy in general. This was done with the intention of validating such a comprehensive system against state-of-the-art commercial and open source baselines on a refined dataset, thereby showcasing its real-world relevance as a secure and trusted means to digitally represent legal documents.

7.2 Principal Contributions and Findings

This thesis includes several novel contributions for the domain of applied OCR, especially in the case of complex and challenging Greek legal documents. The main contribution lies in providing performance benchmark on a large scale for this domain. By performing an extensive benchmarking of well-known proprietary solutions—specifically ABBYY FineReader, Adobe Acrobat, and IRIS—on the one hand, and the widely-used open-source engine Tesseract, on the other, we are presenting the first quantitative benchmarking in this direction with

genuine Greek legal documents. The result is very competitive (we reached a WER of 1.39% with ABBYY FineReader), but leaving Tesseract behind by far (The system provided a WER of 33.01%). Our benchmark creates an empirical performance hierarchy, providing essential data for legal professionals and technology stakeholders that commercial, domain-aware solutions are now fundamental for high-stakes legal work.

In addition to overall comparative performance, we also perform an in-depth error analysis specific to the legal domain which is the first analysis of its kind for Greek. One of the main conclusions was the influence of diacritics (tonos), legalese and arbitrary style on the JURIS character recognition rate. To tackle that mix of printed and printed+handwritten region, a significant contribution of this work is the seamless introduction of a semantic segmentation pipeline. Through the use of a DeepLabv3+ model formulation it was able to infer printed text, handwritten annotations and signatures in a fine-grained manner. This partitioning facilitated focus processing strategy, channeling each type of content to the best-fitting OCR engine and thus reducing the error that resulted from one-size-fits-all manner.

In order to overcome the deficiencies of just using only one OCR engine, this thesis proposed and developed an innovative multi-engine pipeline, referred to as TriOCR. This framework contributes an architectural novelty in terms of the enrichment with outputs of ABBYY FineReader, transformer-based model for handwriting (TrOCR) and Tesseract. At the heart of this system are the customized alignment algorithms that employ sequence matching to identify correspondences among disparate outputs, and a custom confidence scoring system which combines the output from the various engines so as to create a robust consensus. The pipeline facilitates a symbiotic relation between the three read engines, and represents a quality assurance framework, telling the user what to check in each individual read, that is crucial for the legal domain.

Lastly, this study illustrated the importance of post-processing through implementing and incorporating a domain-specific Greek legal lexicon. Due to the fact that the domain-specific vocabulary is usually a substantial challenge for OCR systems even better than the state of the art in OCR, this lexicon was based on a large set of legal documents and applied towards focused error correction. This contribution was able to significantly reduce the frequency of recognition errors for legal entities, procedural terminology and case citations, all of which the original analysis indicated as frequent causes of failure. The effective use of this pure lexicon-based approach demonstrates that the combination of raw OCR with domain-aware

linguistic knowledge is indeed needed to reach the accuracy rates needed for professional digitization of legal documents.

7.3 Discussion and Implications

Moreover, the results of this dissertation add value beyond the mere measurement results as there are various practical implications for applying and promoting the adoption of OCR in the legal domain. Such findings clearly direct the selection of a system, the design of an architecture, and the positioning for an organization that wish to digitize the complex legal documents with high accuracy and in a secure manner.

The most obvious and straightforward application of this work is the quite different performance evident from open-source and commercial OCR for working on Greek legal documents. The main motivation for our work is contained in the experimental results, which leave no doubt that a commercial leading engine such as ABBYY FineReader falls in a completely different reliability class as the open-source Tesseract. This missing piece is not just a matter of quantity, however; it is a qualitative different kind of capacity required when dealing with the domain's challenges such as accurate diacritics preservation, complex layout segmentation or specialized names recognition. For law firms, corporate legal departments, or government offices where document accuracy is vital, our study strongly suggests not only that high-quality commercial-grade OCR software is not a luxury but is an essential part of successful and efficient digitization.

More generally than simply evaluating particular engines, this thesis therefore promotes the strategy of hybrid and multi-engine architecture such as that seen in the implemented TriOCR system. Although ABBYY FineReader was the most accurate single engine, the TriOCR system shows that the consensus based methodology is an effective methodology for quality control. Assemble the outputs of all the engines and scan for their commonalities and differences to produce the hyps of high confidences and low confidences. This involves a smart review process, so humans can concentrate on the very limited fraction where machines are not confident, instead of proofreading full documents. This trust but verify approach is particularly well suited for legal settings because it leads to greater overall reliability, creates a defensible audit trail of the digitizing process, and minimizes the odds of dependence on an individual, potentially flawed, system.

Finally, this work makes a good case for how it's feasible to run high-level OCR systems over private, on-premise infrastructure, hitting straight on to the legal sector's tough privacy and data sovereignty needs. Most advanced AI solutions are cloud-native; this is a non-starter for legal practices dealing with privileged information under regulation such as GDPR and attorney-client privilege. The light-weight modular pipeline proposed in this thesis was made to work efficiently with the usual resource-constrained hardware without reliance on external cloud services. By using off-the-shelf high-performing engine, and focusing on smart scheduling and post-processing instead of training huge models from scratch, the system shows that efficient, high-quality document analysis can be achieved without the need for big computational resources. It confirms a key architectural approach to legal tech: That you build powerful AI-driven tools that respect as non-negotiable that no client data leaves your systems unless you allow it to and remains wholly under your control at all times.

7.4 Limitations of the Current Work

Although this thesis effectively shows that for Greek legal documents, a specialized OCR pipeline can work, we need to be aware of the caveats of this approach and the limitations of the research design and implementation. These limitations restrict the scope of the present investigation and motivate the future directions for research presented in the following section.

The evaluation set is carefully processed for domain relevance and is quantitatively limited. The complete experimental verification was carried out over a collection of 15 Greek legal documents. While those chosen cover a number of the formats and complexities found in the Greek legal system, the sample size involved will not include all the variability of legal documents. The judging of the performance results and error patterns may not be directly transferable to other languages and legal systems with different orthographic rules, formatting conventions, and terms, although occurring there as well. It is therefore that ABBYY FineReader's cross-linguistic accuracy and the error rates that we have outlined are to be taken as a fairly reliable indication of performance in the domain, where the performance may doubtlessly be more widely validated on a larger and more diverse international corpus.

Second, the architecture of the TriOCR system is fundamentally a wrapper and booster, instead of a new recognition model proposed from scratch. Consequently, the performance of

the pipeline is essentially conditioned by and effectively limited to what the third-party OCR engines it utilizes can achieve by themselves. The consensus mechanism and post-processing logic of the system will be able to successfully deal with discrepancy problems and eventually output usable results, but it will still leave irreparable mistakes that all the underlying engines share. However, the upper bound accuracy is restrained by the upper bound proficiency of an individual engine in the ensemble (ABBYY FineReader). Moreover, since the commercial engines are used as "black boxes", this research can only study their output behaviors rather than the internal algorithms, which hinders a thorough study on the causes of the failures for the recognition.

Third, although the system architecture was sensibly devised for mixed-content documents, since printed text and handwritten text are segmented, the results of the experimental evaluation concentrate mainly on the accuracy of printed text. We haven't extensively tested the handwriting recognition pipeline based on a pre-trained TrOCR model to the same extent as the printed text parts. Legal documents may include a wide variety of handwritings, from neat notational annotations to highly cursive signatures and deteriorated marginalia. Furthermore, the whole extensive experiments were carried out considering only printed characters rather than handwritten ones, the accuracy for detecting handwritten samples was not tested in sufficient numbers, so that it is difficult to estimate the effectiveness of the system for this challenging situation. The entire pipeline will ultimately only be as good as its initial segmentation step, and a failure of the DeepLabv3+ model to classify a text region correctly would subsequently pollute the OCR step with sub-optimal online engines, which is a final point of failure that we should subject to additional scrutiny as well.

Finally, the project was built and tested as a standalone command-line driven pipeline in a controlled academic setting. It has not been integrated into working with a real-world Legal Practice Management system, or Document Management System (DMS) or User Acceptance Testing it with Legal Professionals. Because it doesn't have a GUI, this may exclude non-technical users, and its actual effect on legal workflows (efficiency improvements or frictionless usability) is theoretical. Questions on scalability in terms of heavy (concurrent) user loads, integrating smoothly into an existing legal IT infrastructure, and long-term maintainability in a production environment were beyond the scope of this thesis which aimed to demonstrate (technical) feasibility and correctness instead.

7.5 Recommendations for Future Work

Given these fundamentals developed here, there are many potential directions in which the system can be extended and improved, as well as limitations that can be overcome. These suggestions vary from modest enhancements of the current pipeline to ambitious, longer-term research objectives.

First, one main aspect for improvement could be the management of handwritten texts. Although the current model is able to segment and route handwritten parts to a dedicated engine like TrOCR, it is limited by the fact that the pre-trained model is general-purpose. In the future, we intend to fine-tune this transformer-based model on a large, clean dataset of handwriting annotations, signatures, and marginal notes from Greek legal documents. This would vastly enhance its ability to correctly identify peculiar legalese and its diction. At the same time, the DeepLabv3+ segmentation model could be improved by being supplied more challenging document layout data that allows the model to more effectively separate inter-linear notes and overlapping text, leading to a cleaner input for the handwriting recognition engine and thus fewer segmentation-based errors.

Second, the manually curated post-correction stage, which has already proven to be very effective, can morph into a more dynamic and context-sensitive system. Rather than operating with a pre-existing dictionary, a semi-automated process can be created to grow the legal vocabulary by learning from new words detected in high-confidence OCR results, or by corrections suggested during manual review. Next, the existing post-correction approaches may be extended to adopt more sophisticated Natural Language Processing (NLP) models. The inclusion of a BERT model fine-tuned on legal corpora would make the system into a contextual error correction system, which can identify and correct the grammatically correct but semantically incorrect errors that are not possible to catch with a simple lookup.

A future long-term goal is to construct a single end-to-end trainable model for legal document analysis in a more aggressive way. Although IPI BAM has been demonstrated to work well, the performance of the TriOCR pipeline is essentially bounded by its constituent parts. Applications like LayoutLM, Donut, and vision-language transformers aim to jointly model textual contents, layout structure, and semantic relationships from a document image. Such a model can be trained on a huge corpus of legal documents, so it can potentially be a system with better accuracy than the current pipeline, since it understands the whole structure and content of the document. It would be a tacit shift from borrowing and combining tools that

already exist to developing a custom, cutting-edge solution for the legal domain, however, this would demand a large number of annotated data and computational resources.

Moreover, there is a lot of good work to be done on scaling the system to the market, and tuning it to actual production-type workloads. Currently, this proof-of-concept processes documents serially, with an eventual goal to replace this with an intelligently-architected library that can parallelize processing using task queues and multiprocessing libraries. This could dramatically increase throughput for mass digitization. A systematic performance evaluation should also be made on a variety of target edge devices e.g. NVIDIA Jetson and Google Coral TPU. This would also include generating tailored model versions using platform-specific toolkits such as TensorRT and TensorFlow Lite for a better understanding of the speed, accuracy, and energy trade-offs across various hardware profiles.

Third, to close the chasm between a prototype that works and a tool that legal professionals can use in practice, usability and workflow integration become the subjects of future work. The current CLI should be replaced by a friendly Graphical User Interface (GUI) that can be used by non-experienced CDSS users to upload the fascicles, choose between strict and majority consensus runs, and visualize the outcome. This interface would integrate a "human-in-the-loop" mechanism to assist human users at the decoding stage, rendering low-confidence words in a different color and a plain text editor to allow users to manually correct words. The input from such a process would be priceless for successively refining the models of the system. Finally, conducting a formal usability test as a collaboration with a law firm would also be beneficial for us to uncover the items we could miss when we implement the tool in actual practice, as it may enhance findings of the current study regarding whether the design and functionality of the tool correspond with the actual need in terms of legal document handling.

Bibliography

[1] Junmiao Wang. A study of the ocr development history and directions of development. In *Highlights in Science, Engineering and Technology*, volume 72. DRPress, 2023. Accessed June 5, 2025.

[2] S. Mori, C.Y. Suen, and K. Yamamoto. Historical review of ocr research and development. *Proceedings of the IEEE*, 80(7):1029–1058, 1992.

[3] Advances in ai-driven document processing, 2023. Torino Thesis.

[4] Haifeng Wang, Changzai Pan, Xiao Guo, Chunlin Ji, and Ke Deng. From object detection to text detection and recognition: A brief evolution history of optical character recognition. *Wiley Interdisciplinary Reviews: Computational Statistics*, 13(5):e1547, 2021.

[5] Kunyi Li and Lu Cao. A review of object detection techniques. In *2020 5th International Conference on Electromechanical Control Technology and Transportation (ICECTT)*, pages 385–390. IEEE, 2020.

[6] Manisha Vashisht and Brijesh Kumar. A survey paper on object detection methods in image processing. In *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, pages 1–4. IEEE, 2020.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Mirjam Cuper, Corine van Dongen, and Tineke Koster. Unraveling Confidence: Examining Confidence Scores as Proxy for OCR Quality. In Gernot A. Fink, Rajiv Jain,

Koichi Kise, and Richard Zanibbi, editors, *Document Analysis and Recognition - ICDAR 2023*, pages 104–120, Cham, 2023. Springer Nature Switzerland.

[9] V. S. Perifanis, T. Giannakopoulos, and E. Kalampokis. A project for the transformation of greek legal documents into legal open data. `https://www.researchgate.net/publication/373640284`, 2023. Accessed 2025-05-25.

[10] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[12] Naragudem Sarika, Nageswararao Sirisala, and Muni Sekhar Velpuru. Cnn based optical character recognition and applications. In *2021 6th International conference on inventive computation technologies (ICICT)*, pages 666–672. IEEE, 2021.

[13] Amirreza Fateh, Mansoor Fateh, and Vahid Abolghasemi. Enhancing optical character recognition: Efficient techniques for document layout analysis and text line detection. *Engineering Reports*, 6(9):e12832, 2024.

[14] Mingxing Tan and EfficientNet Le Q V. rethinking model scaling for convolutional neural networks. 2019. *arXiv preprint arXiv:1905.11946*, 1905.

[15] Phanthakan Kiatphaisansophon, Dittaya Wanvarie, and Nagul Cooharojananone. Efficient text bounding box identification using mask r-cnn: case of thai documents. *IEEE Access*, 2024.

[16] Alexander Gayer and Vladimir V Arlazarov. Muldt: Multilingual ultra-lightweight document text detection for embedded devices. *IEEE Access*, 2024.

[17] Amir Saba, Rim Hantach, and Mohammed Benslimane. Text detection and recognition from piping and instrumentation diagrams. In *2023 8th International Conference on Image, Vision and Computing (ICIVC)*, pages 711–716. IEEE, 2023.

[18] Anoshor B Paul, G Sai Vikrant, et al. Resilient kannada scene text detection: Craft-yolov8 fusion. In *2024 11th International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 345–350. IEEE, 2024.

[19] Karez Hamad and Mehmet Kaya. A Detailed Analysis of Optical Character Recognition Technology. *International Journal of Applied Mathematics Electronics and Computers*, 4(Special Issue-1):244–249, 2016. Publisher: PLUSBASE AKADEMİ ORGANİZASYON VE DANIŞMANLIK.

[20] Noppol Anakpluek, Watcharakorn Pasanta, Latthawan Chantharasukha, Pattanawong Chokratansombat, Pajaya Kanjanakaew, and Thitirat Siriborvornratanakul. Improved tesseract optical character recognition performance on thai document datasets. *Big Data Research*, 39:100508, 2 2025.

[21] Jieying Li. Research on english automatic recognition system based on ocr technology. In *2023 7th Asian Conference on Artificial Intelligence Technology (ACAIT)*, pages 1061–1066. IEEE, 2023.

[22] Gener Serhan, Dattilo Parker, Gajaria Dhruv, Fusco Alexander, and Akoglu Ali. Gpu-based and streaming-enabled implementation of pre-processing flow towards enhancing optical character recognition accuracy and efficiency. *Cluster Computing*, 26(6):3407–3419, 2023.

[23] Okechukwu Ogochukwu Patience, Eziechina Malachy Amaechi, Onyemachi George, and Onuwa Nnachi Isaac. Enhanced text recognition in images using tesseract ocr within the laravel framework. *Asian Journal of Research in Computer Science*, 17(9):58–69, 2024.

[24] Manan Modi, Arpita Shah, Deep Kothadiya, and Mrugendra Rahevar. Optical character recognition: Comparative analysis of tesseract and textract on diverse datasets. In *2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, pages 913–919. IEEE, 2024.

[25] Ivan Gruber, Lukáš Picek, Miroslav Hlaváč, Petr Neduchal, and Marek Hrúz. Improving handwritten cyrillic ocr by font-based synthetic text generator. In *International Conference on the Dynamics of Information Systems*, pages 102–115. Springer, 2023.

[26] Marc Frincu, Marius E Penteliuc, Simina Frincu, Gheorghe Bran, and Manuela Zanescu. Comparing ml ocr engines on texts from 19 th century written in the romanian transitional script. In *2023 IEEE International Conference on Big Data (BigData)*, pages 1501–1506. IEEE, 2023.

[27] Jarmo Koponen, Keijo Haataja, and Pekka Toivanen. A novel deep learning method for recognizing texts printed with multiple different printing methods. *F1000Research*, 12:427, 04 2023.

[28] Yangyu Lu, Yuru Chen, and Shibiao Zhang. Research on the method of recognizing book titles based on paddle ocr. In *2024 4th International Signal Processing, Communications and Engineering Management Conference (ISPCEM)*, pages 1044–1048. IEEE, 2024.

[29] Rameel Ahmed, Noman Shabbir, Muhammad Wasif Raza, Ayesha Zeb, and Hassan Elahi. Evaluation of model degradation in paddleocr, ultocr, and trocr across baseline and tensorflow lite environments. In *2024 International Conference on Robotics and Automation in Industry (ICRAI)*, pages 1–5. IEEE, 2024.

[30] Attila Biró, Antonio Ignacio Cuesta-Vargas, Jaime Martín-Martín, László Szilágyi, and Sándor Miklós Szilágyi. Synthetized multilanguage ocr using crnn and svtr models for realtime collaborative tools. *Applied Sciences*, 13(7):4419, 2023.

[31] Hai Li and Yang Li. Enhanced training methods for multiple languages. In *Proceedings of the Third DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 52–56, 2023.

[32] Dinh-Truong Do, Minh-Phuong Nguyen, and Le-Minh Nguyen. Enhancing zero-shot multilingual semantic parsing: A framework leveraging large language models for data augmentation and advanced prompting techniques. *Neurocomputing*, 618:129108, 2025.

[33] Phillip Schönfelder, Fynn Stebel, Nikos Andreou, and Markus König. Deep learning-based text detection and recognition on architectural floor plans. *Automation in Construction*, 157:105156, 2024.

[34] JiaWei Xiong and Chun Liu. Design and research of industrial high availability character recognition technology. In *2024 IEEE 6th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, volume 6, pages 1472–1476. IEEE, 2024.

[35] Meharuniza Nazeem, R Anitha, S Navaneeth, et al. Open-source ocr libraries: A comprehensive study for low resource language. In *Proceedings of the 21st International Conference on Natural Language Processing (ICON)*, pages 416–421, 2024.

[36] B Anand Kumar, S Tamilarasan, Ajmeera Kiran, Bagepally Tejaswi, Manikonda Geethika Sree, and Mallikarjun Dasarla. Optical character recognition technology using machine learning. In *2023 International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–4. IEEE, 2023.

[37] M.A.M. Salehudin, S.N. Basah, H. Yazid, K.S. Basaruddin, M.J.A. Safar, M.H. Mat Som, and K.A. Sidek. Analysis of optical character recognition using easyocr under image degradation. *Journal of Physics: Conference Series*, 2641(1):012001, nov 2023.

[38] Pulkit Batra, Nimish Phalnikar, Deepesh Kurmi, Jitendra Tembhurne, Parul Sahare, and Tausif Diwan. Ocr-mrd: performance analysis of different optical character recognition engines for medical report digitization. *International Journal of Information Technology*, 16(1):447–455, 2024.

[39] Sri Charitha Pagadala, Pulletikurthi Nithisha, Pallikonda Rahul, Musiboina Ram Mohan Rao, and Akkineni Haritha. Improving document digitization with machine learning-based ocr. *IJSAT-International Journal on Science and Technology*, 16(1), 2025.

[40] Anand Kumar, Pardeep Singh, and Kusum Lata. Comparative study of different optical character recognition models on handwritten and printed medical reports. In *2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, pages 581–586. IEEE, 2023.

[41] S Ustenko and T Ostapovych. Amazon textract and artficial intelegence system at banking document managment system. *Artificial intelligence*, 1:13–28, 2023.

[42] Yi-Quan Li, Hao-Sen Chang, and Daw-Tung Lin. Large-scale printed chinese character recognition for id cards using deep learning and few samples transfer learning. *Applied Sciences*, 12(2):907, 2022.

[43] Karanrat Thammarak, Prateep Kongkla, Yaowarat Sirisathitkul, and Sarun Intakosum. Comparative analysis of tesseract and google cloud vision for thai vehicle reg-

istration certificate. *International Journal of Electrical and Computer Engineering*, 12(2):1849–1858, 2022.

[44] Zhuo Chen, Fei Yin, Xu-Yao Zhang, Qing Yang, and Cheng-Lin Liu. Multrenets: Multilingual text recognition networks for simultaneous script identification and handwriting recognition. *Pattern Recognition*, 108:107555, 2020.

[45] Sarra Rouabhi, Abdennour Azerine, Redouane Tlemsani, Mokhtar Essaid, and Lhassane Idoumghar. Conv-vit fusion for improved handwritten arabic character classification. *Signal, Image and Video Processing*, 18(Suppl 1):355–372, 2024.

[46] Christian Josefh M Sebastian, Mark William J Rogala, and Charmaine C Paglinawan. Neurological disorder rooted-impaired handwriting to text using neural network image processing for special education applications. In *2024 16th International Conference on Computer and Automation Engineering (ICCAE)*, pages 264–269. IEEE, 2024.

[47] Milind Godase, Chandrani Singh, and Kunal Dhongadi. Text finder application for android. *arXiv preprint arXiv:2311.04579*, 2023.

[48] Chun Hong Lam, Chian Wen Too, Whee Yen Wong, Toong Hai Sam, Beh Hooi Ching, and Hoo Meei Hao. Object detection and optical character recognition for automated malaysia driving license and road tax document details retrieval. In *2024 IEEE 6th Symposium on Computers & Informatics (ISCI)*, pages 236–241. IEEE, 2024.

[49] Arnab Ghosh Chowdhury, Nils Schut, and Martin Atzmueller. A hybrid information extraction approach using transfer learning on richly-structured documents. In *LWDA*, pages 13–25, 2021.

[50] B Aakash and A Srilakshmi. Mage: An efficient deployment of python flask web application to app engine flexible using google cloud platform. In *Inventive Communication and Computational Technologies: Proceedings of ICICCT 2020*, pages 59–68. Springer, 2021.

[51] Dheeraj Chahal, Surya Chaitanya Palepu, and Rekha Singhal. Scalable and cost-effective serverless architecture for information extraction workflows. In *Proceedings of the 2nd Workshop on High Performance Serverless Computing*, pages 15–23, 2022.

[52] Iago Lourenço Correa, Paulo Lilles Jorge Drews, and Ricardo Nagel Rodrigues. Combination of optical character recognition engines for documents containing sparse text and alphanumeric codes. In *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 299–306. IEEE, 2021.

[53] Andrey Shapenko, Vladimir Korovkin, and Benoit Leleux. ABBYY: the digitization of language and text. *Emerald Emerging Markets Case Studies*, 8(2):1–26, January 2018. Publisher: Emerald Publishing Limited.

[54] ABBYY Production LLC. *ABBYY FineReader Engine - Performance Guide*, 2017. Accessed June 4, 2025.

[55] Ahmad P. Tafti, Ahmadreza Baghaie, Mehdi Assefi, Hamid R. Arabnia, Zeyun Yu, and Peggy Peissig. Ocr as a service: An experimental evaluation of google docs ocr, tesseract, abbyy finereader, and transym. *ResearchGate*, 2016. Accessed June 5, 2025.

[56] Made Windu Antara Kesiman, Jean-Christophe Burie, and Jean-Marc Ogier. A complete scheme of spatially categorized glyph recognition for the transliteration of balinese palm leaf manuscripts. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 125–130, 2017.

[57] IMPACT Centre of Competence in Digitisation. ocrevalUAtion Tool. `https://github.com/impactcentre/ocrevalUAtion`, 2019. Accessed June 5, 2025.

[58] Mirjam Cuper, Corine van Dongen, and Tineke Koster. Unraveling confidence: Examining confidence scores as proxy for ocr quality. In Gernot A. Fink, Rajiv Jain, Koichi Kise, and Richard Zanibbi, editors, *Document Analysis and Recognition - ICDAR 2023*, pages 104–120, Cham, 2023. Springer Nature Switzerland.

[59] Puneeth Prakash, Sharath Kumar Yeliyur Hanumanthaiah, and Somashekhar Bannur Mayigowda. Crnn model for text detection and classification from natural scenes. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 13(1):839, 2024.

[60] Liao Huilian, Du Xingwei, He Luhang, Wang Shanlei, Yao Meng, and Zou Hongbo. A terminal tube text detection and recognition method based on improved yolov7-tiny and crnn. *IEEE Access*, 2024.

[61] Junjie Ma, Shuqiang Huang, Jing Zhang, Wei Sun, Longyue Li, Hao Zhou, Yong Ma, and Hong Ye. Enhancing corporate data security: A mobilenetv3-based approach for complex scene payment numeric text data recognition. In *2024 5th International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, pages 729–736. IEEE, 2024.

[62] George Retsinas, Konstantina Nikolaidou, and Giorgos Sfikas. Enhancing crnn htr architectures with transformer blocks. In *International Conference on Document Analysis and Recognition*, pages 425–440. Springer, 2024.

[63] Rajendra Paudyal and Dhiraj Pyakurel. Enhancing the efficiency of deep learning models for handwritten text recognition by utilizing meta-learning optimization techniques. *Journal of Advanced College of Engineering and Management*, 9:1–13, 2024.

[64] Dwi Ahmad Dzulhijjah, Kusrini Kusrini, and Kumara Ari Yuana. Comparative analysis of rnn, lstm, bi-lstm performance for location and time entity recognition in forest fire texts. In *2024 2nd International Conference on Software Engineering and Information Technology (ICoSEIT)*, pages 181–186. IEEE, 2024.

[65] Sohail Zia, Muhammad Azhar, Bumshik Lee, Adnan Tahir, Javed Ferzund, Fozia Murtaza, and Moazam Ali. Recognition of printed urdu script in nastaleeq font by using cnn-bigru-gru based encoder-decoder framework. *Intelligent Systems with Applications*, 18:200194, 2023.

[66] Poluru Eswaraiah and Hussain Syed. A hybrid deep learning gru based approach for text classification using word embedding. *EAI Endorsed Transactions on Internet of Things*, 10, 2024.

[67] Le Tran Anh Dang, Vong Duong Ngoc, Pham Cung Le Thien Vu, Nguyen Nhat Truong, Pham The Bao, and Tan Dat Trinh. Vietnam vehicle number recognition based on an improved crnn with attention mechanism. *International Journal of Intelligent Transportation Systems Research*, 22(2):374–389, 2024.

[68] Guo Zhijun, Luo Weiming, Chen Qiujie, and Zou Hongbo. Terminal strip detection and recognition based on improved yolov7-tiny and mah-crnn+ ctc models. *Frontiers in Energy Research*, 12:1345574, 2024.

[69] Tayyab Nasir and Muhammad Kamran Malik. Efficient crnn: Towards end-to-end low resource urdu text recognition using depthwise separable convolutions and gated recurrent units. *Information Processing & Management*, 61(1):103544, 2024.

[70] Le Zou, Zhihuang He, Kai Wang, Zhize Wu, Yifan Wang, Guanhong Zhang, and Xiaofeng Wang. Text recognition model based on multi-scale fusion crnn. *Sensors*, 23(16):7034, 2023.

[71] Shengying Yang, Lifeng Xie, Xiaoxiao Ran, Jingsheng Lei, and Xiaohong Qian. Pragmatic degradation learning for scene text image super-resolution with data-training strategy. *Knowledge-Based Systems*, 285:111349, 2024.

[72] Qin Shi, Yu Zhu, Yatong Liu, Jiongyao Ye, and Dawei Yang. Perceiving multiple representations for scene text image super-resolution guided by text recognizer. *Engineering Applications of Artificial Intelligence*, 124:106551, 2023.

[73] Xin Ying, Raja Kumar Murugesan, Siva Raja Sindiramutty, Goh Wei Wei, Sumathi Balakrishnan, Devender Kumar, and Sahil Verma. Scene text recognition using deep learning techniques. In *2024 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, pages 1–9. IEEE, 2024.

[74] Yiyi Liu, Yuxin Wang, and Hongjian Shi. A convolutional recurrent neural-network-based machine learning for scene text recognition application. *Symmetry*, 15(4):849, 2023.

[75] Jinwoo Kim, Daeho Kim, SangHyun Lee, and Seokho Chi. Hybrid dnn training using both synthetic and real construction images to overcome training data shortage. *Automation in Construction*, 149:104771, 2023.

[76] Ishfaq Hussain Rather and Sushil Kumar. Generative adversarial network based synthetic data training model for lightweight convolutional neural networks. *Multimedia Tools and Applications*, 83(2):6249–6271, 2024.

[77] Yaganteeswarudu Akkem, Saroj Kumar Biswas, and Aruna Varanasi. A comprehensive review of synthetic data generation in smart farming by using variational autoencoder and generative adversarial network. *Engineering Applications of Artificial Intelligence*, 131:107881, 2024.

[78] Hideaki Yajima, Chee Siang Leow, and Hiromitsu Nishizaki. Text detection and style classification from images using vision transformer and transformer decoder. In *2024 IEEE 13th Global Conference on Consumer Electronics (GCCE)*, pages 619–623. IEEE, 10 2024.

[79] Chaoya Jiang, Haiyang Xu, Wei Ye, Qinghao Ye, Chenliang Li, Ming Yan, Bin Bi, Shikun Zhang, Fei Huang, and Ji Zhang. Copa : Efficient vision-language pre-training through collaborative object- and patch-text alignment. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 4480–4491. ACM, 10 2023.

[80] Syed Muhammad Ahmed Hassan Shah, Muhammad Qasim Khan, Yazeed Yasin Ghadi, Sana Ullah Jan, Olfa Mzoughi, and Monia Hamdi. A hybrid neuro-fuzzy approach for heterogeneous patch encoding in vits using contrastive embeddings and deep knowledge dispersion. *IEEE Access*, 11:83171–83186, 2023.

[81] Lei Wang, Xue song Tang, and Kuangrong Hao. Gfpe-vit: vision transformer with geometric-fractal-based position encoding. *The Visual Computer*, 41:1021–1036, 1 2025.

[82] Yuting Li, Dexiong Chen, Tinglong Tang, and Xi Shen. Htr-vt: Handwritten text recognition with vision transformer. *Pattern Recognition*, 158:110967, 2 2025.

[83] Ran Shao, Xiao-Jun Bi, and Zheng Chen. Hybrid vit-cnn network for fine-grained image classification. *IEEE Signal Processing Letters*, 31:1109–1113, 2024.

[84] Feng Zhao, Junjie Zhang, Zhe Meng, Hanqiang Liu, Zhenhui Chang, and Jiulun Fan. Multiple vision architectures-based hybrid network for hyperspectral image classification. *Expert Systems with Applications*, 234:121032, 12 2023.

[85] Jeff Yang, Huynh Vu The, and Hai Luu Tuan. Light-weight multi-modality feature fusion network for visually-rich document understanding. In Elisa H. Barney Smith, Marcus Liwicki, and Liangrui Peng, editors, *Document Analysis and Recognition - ICDAR 2024*, pages 191–207, Cham, 2024. Springer Nature Switzerland.

[86] Ibrahim Batuhan Akkaya, Senthilkumar S. Kathiresan, Elahe Arani, and Bahram Zonooz. Enhancing performance of vision transformers on small datasets through local inductive bias incorporation. *Pattern Recognition*, 153:110510, 9 2024.

[87] Jin Gao, Shubo Lin, Shaoru Wang, Yutong Kou, Zeming Li, Liang Li, Congxuan Zhang, Xiaoqin Zhang, Yizheng Wang, and Weiming Hu. An experimental study on exploring strong lightweight vision transformers via masked image modeling pre-training. *International Journal of Computer Vision*, 2 2025.

[88] Srijan Das, Tanmay Jain, Dominick Reilly, Pranav Balaji, Soumyajit Karmakar, Shyam Marjit, Xiang Li, Abhijit Das, and Michael S. Ryoo. Limited data, unlimited potential: A study on vits augmented by masked autoencoders. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6864–6874. IEEE, 1 2024.

[89] CHEN HUAN, WENTAO WEI, and Ping Yao. Train vit on small dataset with translation perceptibility. In *34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, November 20-24, 2023*. BMVA, 2023.

[90] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *CoRR*, abs/1708.04896, 2017.

[91] Bum Jun Kim and Sang Woo Kim. Configuring data augmentations to reduce variance shift in positional embedding of vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 17840–17849, 2025.

[92] Mehrdad Noori, Milad Cheraghalikhani, Ali Bahri, Gustavo A Vargas Hakim, David Osowiechi, Ismail Ben Ayed, and Christian Desrosiers. Tfs-vit: Token-level feature stylization for domain generalization. *Pattern Recognition*, 149:110213, 2024.

[93] Cheng Da, Chuwei Luo, Qi Zheng, and Cong Yao. Vision grid transformer for document layout analysis. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19405–19415. IEEE, 10 2023.

[94] Thibault Douzon, Stefan Duffner, Christophe Garcia, and Jérémy Espinas. *Long-Range Transformer Architectures for Document Understanding*, page 47–64. Springer Nature Switzerland, 2023.

[95] Gang Yang and Hongzhe Xu. Complex-valued relative positional encodings for transformer. In *2023 3rd International Conference on Neural Networks, Information and Communication Engineering (NNICE)*, pages 373–379. IEEE, 2 2023.

[96] Xueming Yan, Zhihang Fang, and Yaochu Jin. An adaptive n-gram transformer for multi-scale scene text recognition. *Knowledge-Based Systems*, 280:110964, 11 2023.

[97] Ke Li, Di Wang, Gang Liu, Wenxuan Zhu, Haodi Zhong, and Quan Wang. Diagswin: A multi-scale vision transformer with diagonal-shaped windows for object detection and segmentation. *Neural Networks*, 180:106653, 12 2024.

[98] Gener Serhan, Dattilo Parker, Gajaria Dhruv, Fusco Alexander, and Akoglu Ali. Gpu-based and streaming-enabled implementation of pre-processing flow towards enhancing optical character recognition accuracy and efficiency. *Cluster Computing*, 26:3407–3419, 12 2023.

[99] Weiduo Chen, Xiaoshe Dong, Xinhang Chen, Song Liu, Qin Xia, and Qiang Wang. pommdnn: Performance optimal gpu memory management for deep neural network training. *Future Generation Computer Systems*, 152:160–169, 3 2024.

[100] Shuibing He, Ping Chen, Shuaiben Chen, Zheng Li, Siling Yang, Weijian Chen, and Lidan Shou. Home: A holistic gpu memory management framework for deep learning. *IEEE Transactions on Computers*, pages 1–13, 2022.

[101] Han Zhang, Hui Wang, and Ruifeng Xu. Incremental pre-training from smaller language models. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 36–44. Association for Computational Linguistics, 2024.

[102] Qinlu He, Fan Zhang, Genqing Bian, Weiqi Zhang, and Zhen Li. Design and realization of hybrid resource management system for heterogeneous cluster. *Cluster Computing*, 27:6119–6144, 8 2024.

[103] Hoda Sedighi, Daniel Gehberger, Amin Ebrahimzadeh, Fetahi Wuhib, and Roch H. Glitho. Efficient dynamic resource management for spatial multitasking gpus. *IEEE Transactions on Cloud Computing*, 13:99–117, 1 2025.

[104] Juan Fang, Sheng Lin, Huijing Yang, Yixiang Xu, and Xing Su. A perceptual and predictive batch-processing memory scheduling strategy for a cpu-gpu heterogeneous system. *Frontiers of Information Technology & Electronic Engineering*, 24:994–1006, 7 2023.

[105] Marion Dörrich, Mingcheng Fan, and Andreas M. Kist. Impact of mixed precision techniques on training and inference efficiency of deep neural networks. *IEEE Access*, 11:57627–57634, 2023.

[106] Hao Li, Yuzhu Wang, Yan Hong, Fei Li, and Xiaohui Ji. Layered mixed-precision training: A new training method for large-scale ai models. *Journal of King Saud University - Computer and Information Sciences*, 35:101656, 9 2023.

[107] Uchechukwu Leo Udeji and Martin Margala. Segmentai: A neural net framework for optimized multiclass image segmentation via fpga. In *2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 421–426. IEEE, 7 2024.

[108] Joel Hayford, Jacob Goldman-Wetzler, Eric Wang, and Lu Lu. Speeding up and reducing memory usage for scientific machine learning via mixed precision. *Computer Methods in Applied Mechanics and Engineering*, 428:117093, 2024.

[109] Wenhao Dai, Ziyi Jia, Yuesi Bai, and Qingxiao Sun. Convergence-aware operator-wise mixed-precision training. *CCF Transactions on High Performance Computing*, 7:43–57, 2 2025.

[110] Renbo Tu, Colin White, Jean Kossaifi, Boris Bonev, Nikola Kovachki, Gennady Pekhimenko, Kamyar Azizzadenesheli, and Anima Anandkumar. Guaranteed approximation bounds for mixed-precision neural operators. *arXiv preprint arXiv:2307.15034*, 2023.

[111] Zhuo Tang, Zhanfei Xiao, Li Yang, Kailin He, and Kenli Li. A network load perception based task scheduler for parallel distributed data processing systems. *IEEE Transactions on Cloud Computing*, 11:1352–1364, 4 2023.

[112] M. Rosyad Ahsanul Wafa, Muhaimin Toh-arlim, Sritrusta Sukaridhoto, Rizqi Putri Nourma Budiarti, and Agus Prayudi. Optimizing optical character recognition accuracy in livestock weighting systems through distributed intelligent computer vision. In *2024 IEEE International Symposium on Consumer Technology (ISCT)*, pages 367–371. IEEE, 8 2024.

[113] Zhigang Wang, Yilei Tu, Ning Wang, Lixin Gao, Jie Nie, Zhiqiang Wei, Yu Gu, and Ge Yu. Fsp: Towards flexible synchronous parallel frameworks for distributed ma-

chine learning. *IEEE Transactions on Parallel and Distributed Systems*, 34:687–703, 2 2023.

[114] Vijayakumar H. Bhajantri. A hybrid parallel techniques for large-scale text classification with symmetric admm and 1-factorization. *Fuel Cells Bulletin*, pages 238–252, 12 2024.

[115] Le Zhou. Longt5-mulla: Longt5 with multi-level local attention for a longer sequence. *IEEE Access*, 11:138433–138444, 2023.

[116] Tiangang Li, Shi Ying, Yishi Zhao, and Jianga Shang. Batch jobs load balancing scheduling in cloud computing using distributional reinforcement learning. *IEEE Transactions on Parallel and Distributed Systems*, 35:169–185, 1 2024.

[117] Jinming Wang, Shaobo Li, Xingxing Zhang, Keyu Zhu, Cankun Xie, and Fengbin Wu. Deep reinforcement learning task scheduling method for real-time performance awareness. *IEEE Access*, 13:31385–31400, 2025.

[118] Junnan Liu, Yifan Liu, and Yongkang Ding. Research and optimization of task scheduling algorithm based on heterogeneous multi-core processor. *Cluster Computing*, 27:13435–13453, 12 2024.

[119] Min Wang, Jiawang Chen, Haoyuan Wang, Ziyi Gao, Weihao Bian, and Sibo Qiao. An enhanced list scheduling algorithm for heterogeneous computing using an optimized predictive cost matrix. *Future Generation Computer Systems*, 166:107733, 5 2025.

[120] Sri Charitha Pagadala, Pulletikurthi Nithisha, Pallikonda Rahul, Musiboina Ram Mohan Rao, and Dr. Akkineni. Haritha. Improving document digitization with machine learning-based ocr. *International Journal on Science and Technology*, 16, 2 2025.

[121] Khadija Awan, Sumbal Khan, Shahab Haider, Noreen Khan, Zulfiqar Ali, and Robertas Damaševicius. Triopen: A novel model to prioritize responsive flows enabling enhanced multimedia communication on the internet. *Multimedia Tools and Applications*, 10 2024.

[122] Rakandhiya Daanii Rachmanto, Ahmad Naufal Labiib Nabhaan, and Arief Setyanto. Deep learning model compression techniques performance on edge devices. *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, 23:569–582, 6 2024.

[123] Bowen Yao, Liansheng Liu, Yu Peng, and Xiyuan Peng. Intelligent measurement on edge devices using hardware memory-aware joint compression enabled neural networks. *IEEE Transactions on Instrumentation and Measurement*, 73:1–13, 2024.

[124] Fazal Muhammad Ali Khan, Hatem Abou-Zeid, and Syed Ali Hassan. Deep compression for efficient and accelerated over-the-air federated learning. *IEEE Internet of Things Journal*, 11:25802–25817, 8 2024.

[125] Siqi Li, Jun Chen, Shanqi Liu, Chengrui Zhu, Guanzhong Tian, and Yong Liu. Mcmc: Multi-constrained model compression via one-stage envelope reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36:3410–3422, 2 2025.

[126] Xinyu Liu, Tao Wang, Jiaming Yang, Chenwei Tang, and Jiancheng Lv. Mpq-yolo: Ultra low mixed-precision quantization of yolo for edge devices deployment. *Neurocomputing*, 574:127210, 3 2024.

[127] Rameel Ahmed, Noman Shabbir, Muhammad Wasif Raza, Ayesha Zeb, and Hassan Elahi. Evaluation of model degradation in paddleocr, ultocr, and trocr across baseline and tensorflow lite environments. In *2024 International Conference on Robotics and Automation in Industry (ICRAI)*, pages 1–5. IEEE, 12 2024.

[128] G. Janaki and D. Umanandhini. Federated learning approaches for decentralized data processing in edge computing. In *2024 5th International Conference on Smart Electronics and Communication (ICOSEC)*, pages 513–519. IEEE, 9 2024.

[129] Charuka Herath, Xiaolan Liu, Sangarapillai Lambotharan, and Yogachandran Rahulamathavan. Enhancing federated learning convergence with dynamic data queue and data-entropy-driven participant selection. *IEEE Internet of Things Journal*, 12:6646–6658, 3 2025.

[130] Neha Singh and Mainak Adhikari. Selffed: Self-adaptive federated learning with non-iid data on heterogeneous edge devices for bias mitigation and enhance training efficiency. *Information Fusion*, 118:102932, 6 2025.

[131] Andrea Silvi, Andrea Rizzardi, Debora Caldarola, Barbara Caputo, and Marco Ciccone. Accelerating federated learning via sequential training of grouped heterogeneous clients. *IEEE Access*, 12:57043–57058, 2024.

[132] Yongheng Deng, Feng Lyu, Tengxi Xia, Yuezhi Zhou, Yaoxue Zhang, Ju Ren, and Yuanyuan Yang. A communication-efficient hierarchical federated learning framework via shaping data distribution at edge. *IEEE/ACM Transactions on Networking*, 32:2600–2615, 6 2024.

[133] Tianao Xiang, Yuanguo Bi, Xiangyi Chen, Yuan Liu, Boyang Wang, Xuemin Shen, and Xingwei Wang. Federated learning with dynamic epoch adjustment and collaborative training in mobile edge computing. *IEEE Transactions on Mobile Computing*, 23:4092–4106, 5 2024.

[134] Mingda Hu, Jingjing Zhang, Xiong Wang, Shengyun Liu, and Zheng Lin. Accelerating federated learning with model segmentation for edge networks. *IEEE Transactions on Green Communications and Networking*, 9:242–254, 3 2025.

[135] Rituparna Saha, Sudip Misra, Aishwariya Chakraborty, Chandranath Chatterjee, and Pallav Kumar Deb. Data-centric client selection for federated learning over distributed edge networks. *IEEE Transactions on Parallel and Distributed Systems*, 34:675–686, 2 2023.

[136] Tusar Kanti Mishra, Manjur Kolhar, Soumya Ranjan Mishra, Hitesh Mohapatra, Fadi Al-Turjman, and Amiya Kumar Rath. Local features-based evidence glossary for generic recognition of handwritten characters. *Neural Computing and Applications*, 36:685–695, 1 2024.

[137] Enrique Vidal, Alejandro H. Toselli, Antonio Ríos-Vila, and Jorge Calvo-Zaragoza. End-to-end page-level assessment of handwritten text recognition. *Pattern Recognition*, 142:109695, 10 2023.

[138] Khulood Gaashan and Maram Bani Younes. Deep learning-based arabic optical character recognition: A new comprehensive dataset at character and word levels. In *2024 15th International Conference on Information and Communication Systems (ICICS)*, pages 1–6. IEEE, 8 2024.

[139] Buddaraju Revathi, M. V. D. Prasad, and Naveen Kishore Gattim. Computationally efficient resnet based telugu handwritten text detection. *Bulletin of Electrical Engineering and Informatics*, 13:4115–4123, 12 2024.

[140] Mei-Yuh Hwang, Yangyang Shi, Ankit Ramchandani, Guan Pang, Praveen Krishnan, Lucas Kabela, Frank Seide, Samyak Datta, and Jun Liu. Disgo: Automatic end-to-end evaluation for scene text ocr. *arXiv preprint arXiv:2308.13173*, 2023.

[141] Aram Lee, HongYeon Yu, and Gihyeon Min. An algorithm of line segmentation and reading order sorting based on adjacent character detection: A post-processing of ocr for digitization of chinese historical texts. *Journal of Cultural Heritage*, 67:80–91, 5 2024.

[142] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67:220102, 12 2024.

[143] Ling Fu, Biao Yang, Zhebin Kuang, Jiajun Song, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, and Mingxin Huang. Ocrbench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. *arXiv preprint arXiv:2501.00321*, 2024.

[144] Zhibo Yang, Jun Tang, Zhaohai Li, Pengfei Wang, Jianqiang Wan, Humen Zhong, Xuejing Liu, Mingkun Yang, Peng Wang, and Yuliang Liu. Cc-ocr: A comprehensive and challenging ocr benchmark for evaluating large multimodal models in literacy. *arXiv preprint arXiv:2412.02210*, 2024.

[145] Ahmed Heakl, Abdullah Sohail, Mukul Ranjan, Rania Hossam, Ghazi Ahmed, Mohamed El-Geish, Omar Maher, Zhiqiang Shen, Fahad Khan, and Salman Khan. Kitab-bench: A comprehensive multi-domain benchmark for arabic ocr and document understanding. *arXiv preprint arXiv:2502.14949*, 2025.

[146] Birhanu Hailu Belay, Isabelle Guyon, Tadele Mengiste, Bezawork Tilahun, Marcus Liwicki, Tesfa Tegegne, and Romain Egele. A historical handwritten dataset for ethiopic ocr with baseline models and human-level performance. In Elisa H. Barney Smith, Marcus Liwicki, and Liangrui Peng, editors, *Document Analysis and Recognition - ICDAR 2024*, pages 23–38, Cham, 2024. Springer Nature Switzerland.

[147] Jenna Kefeli and Nicholas Tatonetti. Tcga-reports: A machine-readable pathology report resource for benchmarking text-based ai models. *Patterns*, 5:100933, 3 2024.

[148] Miao Rang, Zhenni Bi, Chuanjian Liu, Yunhe Wang, and Kai Han. An empirical study of scaling law for ocr. *arXiv preprint arXiv:2401.00028*, 2023.

[149] Sankalp Nagaonkar, Augustya Sharma, Ashish Choithani, and Ashutosh Trivedi. Benchmarking vision-language models on optical character recognition in dynamic video environments. *arXiv preprint arXiv:2502.06445*, 2025.

[150] Arthur Hemmer, Mickaël Coustaty, Nicola Bartolo, and Jean-Marc Ogier. Confidence-aware document ocr error detection, 2024.

[151] Rina Buoy, Masakazu Iwamura, Sovila Srun, and Koichi Kise. Toward a low-resource non-latin-complete baseline: An exploration of khmer optical character recognition. *IEEE Access*, 11:128044–128060, 2023.

[152] Milind Agarwal and Antonios Anastasopoulos. A concise survey of ocr for low-resource languages. In *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)*, pages 88–102. Association for Computational Linguistics, 2024.

[153] Muhammad Abdullah Sohail, Salaar Masood, and Hamza Iqbal. Deciphering the underserved: Benchmarking llm ocr for low-resource scripts. *arXiv preprint arXiv:2412.16119*, 2024.

[154] Anirudha Ghosh, Debaditya Barman, Abu Sufian, and Ibrahim A. Hameed. Advancing optical character recognition for low-resource scripts: A siamese meta-learning approach with psn framework. *IEEE Access*, 12:189651–189666, 2024.

[155] Prawaal Sharma, Poonam Goyal, Vidisha Sharma, and Navneet Goyal. Voltage: A versatile contrastive learning based ocr methodology for ultra low-resource scripts through auto glyph feature extraction. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 881–899, 2024.

[156] Fatemeh Asadi-Zeydabadi, Elham Shabaninia, Hossein Nezamabadi-Pour, and Melika Shojaee. Farsi optical character recognition using a transformer-based model. In *2023 13th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 293–299. IEEE, 11 2023.

[157] Harshvivek Kashid and Pushpak Bhattacharyya. Roundtripocr: A data generation technique for enhancing post-ocr error correction in low-resource devanagari languages. *arXiv preprint arXiv:2412.15248*, 2024.

[158] Vishnuvardhan Atmakuri and M. Dhanalakshmi. Improving text recognition in natural scenes using optimized ocr techniques. In *2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, pages 943–950. IEEE, 12 2024.

[159] Abdul Majid, Qinbo, Dil Nawaz Hakro, and Saba Brahmani. Thinning chinese, korean, japanese and thai script for segmentation-free ocrs. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pages 116–121, 1 2024.

[160] Musa Dildar Ahmed Cheema, Mohammad Daniyal Shaiq, Farhaan Mirza, Ali Kamal, and M. Asif Naeem. Adapting multilingual vision language transformers for low-resource urdu optical character recognition (ocr). *PeerJ Computer Science*, 10:e1964, 4 2024.

[161] Shuguang Xiong, Xiaoyang Chen, and Huitao Zhang. Deep learning-based multifunctional end-to-end model for optical character classification and denoising. *Journal of Computational Methods in Engineering Applications*, pages 1–13, 11 2023.

[162] Giordano Cicchetti and Danilo Comminiello. Naf-dpm: A nonlinear activation-free diffusion probabilistic model for document enhancement. *arXiv preprint arXiv:2404.05669*, 2024.

[163] Katyani Singh, Ganesh Tata, Eric Van Oeveren, and Nilanjan Ray. Unpaired document image denoising for ocr using bilstm enhanced cyclegan. *International Journal on Document Analysis and Recognition (IJDAR)*, 28:207–224, 6 2025.

[164] Jiahao Quan, Hailing Wang, Chunwei Wu, and Guitao Cao. Dle: Document illumination correction with dynamic light estimation. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3701–3707. IEEE, 10 2024.

[165] Ahana Kundu and Ujjwal Bhattacharya. Yolo assisted a* algorithm for robust line segmentation of degraded document images. In Elisa H. Barney Smith, Marcus Liwicki,

and Liangrui Peng, editors, *Document Analysis and Recognition - ICDAR 2024*, pages 407–424, Cham, 2024. Springer Nature Switzerland.

[166] Percy Maldonado-Quispe and Helio Pedrini. An effective approach to text detection and recognition in degraded historical documents. In Ruber Hernández-García, Ricardo J. Barrientos, and Sergio A. Velastin, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 256–269, Cham, 2025. Springer Nature Switzerland.

[167] Angel Beshirov, Milena Dobreva, Dimitar Dimitrov, Momchil Hardalov, Ivan Koychev, and Preslav Nakov. Post-ocr text correction for bulgarian historical documents. *International Journal on Digital Libraries*, 26:4, 2025.

[168] Edwin Roi Casas-Huamanta, Lloy Pinedo, Enrique Alejandro Barbachán-Ruales, Angel Cardenas-García, Luis Alberth Rossel-Bernedo, and Jose Gabriel Seijas-Díaz. Optical character recognition system with natural language processing for data recovery on scanned old academic card reports. *Acta Scientiarum. Technology*, 47:e69814, 12 2024.

[169] Arnab Poddar, Soumyadeep Dey, Pratik Jawanpuria, Jayanta Mukhopadhyay, and Prabir Kumar Biswas. Tbm-gan: Synthetic document generation with degraded background. In Gernot A. Fink, Rajiv Jain, Koichi Kise, and Richard Zanibbi, editors, *Document Analysis and Recognition - ICDAR 2023*, pages 366–383, Cham, 2023. Springer Nature Switzerland.

[170] Yilin Chen and Zhan Shi. A printed formula recognition method based on the ”encoder-decoder” deep learning model. In *2023 4th International Conference on Computer Engineering and Intelligent Control (ICCEIC)*, pages 444–448. IEEE, 10 2023.

[171] Sofi.A. Francis and M. Sangeetha. A comparison study on optical character recognition models in mathematical equations and in any language. *Results in Control and Optimization*, 18:100532, 3 2025.

[172] Alexei Kaltchenko. Assessing gpt model uncertainty in mathematical ocr tasks via entropy analysis. *International Journal of Information Technology*, pages 1–10, 2025.

[173] Nasuha Iskandar, Wei Jen Chew, and Swee King Phang. The application of image processing for conversion of handwritten mathematical expression. *Journal of Physics: Conference Series*, 2523:012014, 7 2023.

[174] Nan Zhang, Connor Heaton, Sean Timothy Okonsky, Prasenjit Mitra, and Hilal Ezgi Toraman. Peace: A chemistry-oriented dataset for optical character recognition on scientific documents. *arXiv preprint arXiv:2403.15724*, 2024.

[175] Yogita Hande and Ritika Pandey. Neural network grading: Automated evaluation of theoretical, mathematical, and diagrammatic responses. In *2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*, pages 1–6. IEEE, 8 2023.

[176] Yu Sun, Dongzhan Zhou, Chen Lin, Conghui He, Wanli Ouyang, and Han-Sen Zhong. Locr: Location-guided transformer for optical character recognition. *arXiv preprint arXiv:2403.02127*, 2024.

[177] Yizhi Zou, Han Cao, Xu Cheng, and Lu Yang. A deep learning-based automatic data acquisition system for medical monitors. In *2024 14th International Conference on Information Science and Technology (ICIST)*, pages 200–207. IEEE, 12 2024.

[178] Prof. Rajesh Nasare, Ayush Namdeo, Lokesh Bagmare, Omkant Shende, Anshini Kumbhare, Nishank Hedaoo, and Soham Sahare. Automatic signature and photo detection from the uploaded document using ai and ocr. *International Journal of Advanced Research in Science, Communication and Technology*, pages 415–422, 4 2024.

[179] Jamshed Memon, Maira Sami, Rizwan Ahmed Khan, and Mueen Uddin. Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). *IEEE Access*, 8:142642–142668, 2020.

[180] Disha Agarwal, Jeevan J, R Karthick Manikandan, N R Ramith, and Vandana M L. Advanced automated document processing using optical character recognition (ocr). In *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, pages 1–5. IEEE, 4 2024.

[181] Nataliya SHAKHOVSKA and Andrii SHEBEKO. Development of the architecture of document optical character recognition system. *Herald of Khmelnytskyi National University. Technical sciences*, 309:50–54, 5 2022.

[182] S. Iwin Thanakumar Joseph. Advanced digital image processing technique based optical character recognition of scanned document. *Journal of Innovative Image Processing*, 4:195–205, 10 2022.

[183] Amirreza Fateh, Mansoor Fateh, and Vahid Abolghasemi. Enhancing optical character recognition: Efficient techniques for document layout analysis and text line detection. *Engineering Reports*, 6, 9 2024.

[184] Chun Hong Lam, Chian Wen Too, Whee Yen Wong, Toong Hai Sam, Beh Hooi Ching, and Hoo Meei Hao. Object detection and optical character recognition for automated malaysia driving license and road tax document details retrieval. In *2024 IEEE 6th Symposium on Computers & Informatics (ISCI)*, pages 236–241. IEEE, 8 2024.

[185] Thi Tuyet Hai Nguyen, Adam Jatowt, Mickael Coustaty, and Antoine Doucet. Survey of post-ocr processing approaches. *ACM Computing Surveys*, 54:1–37, 7 2022.

[186] Sukhjinder Singh and Naresh Kumar Garg. Review of optical devanagari character recognition techniques. In Suresh Chandra Satapathy, Vikrant Bhateja, B. Janakiramaiah, and Yen-Wei Chen, editors, *Intelligent System Design*, pages 97–106, Singapore, 2021. Springer Singapore.

[187] Supriya Mahadevkar, Shruti Patil, and Ketan Kotecha. Enhancement of handwritten text recognition using ai-based hybrid approach. *MethodsX*, 12:102654, 6 2024.

[188] Husam Ahmad Alhamad, Mohammad Shehab, Mohd Khaled Y. Shambour, Muhannad A. Abu-Hashem, Ala Abuthawabeh, Hussain Al-Aqrabi, Mohammad Sh. Daoud, and Fatima B. Shannaq. Handwritten recognition techniques: A comprehensive review. *Symmetry*, 16:681, 6 2024.

[189] Akm Shahariar Azad Rabby, Hasmot Ali, Md. Majedul Islam, and Fuad Rahman. Versatile bengali ocr: Document analysis technique for varied document styles and content. In *2023 IEEE International Conference on Big Data (BigData)*, pages 1965–1969. IEEE, 12 2023.

[190] Harichandana Gonuguntla, Kurra Meghana, Thode Sai Prajwal, Koundinya NVSS, Kolli Nethre Sai, and Chirag Jain. Evaluating modern information extraction techniques for complex document structures. In *2024 International Conference on Electrical, Computer and Energy Technologies (ICECET*, pages 1–5. IEEE, 7 2024.

[191] Xusheng Liang, Abbas Cheddad, and Johan Hall. Comparative study of layout analysis of tabulated historical documents. *Big Data Research*, 24:100195, 5 2021.

[192] Viet Anh Pham. Improved ocr quality for smart scanned document management system. *Journal of Science and Technique*, 9, 5 2022.

[193] Tsanta Christelle Nadège Ralaibozaka, Maminiaina Alphonse Rafidison, and Hajasoa Malalatiana Ramafiarisona. Contribution to the authenticity of digitized handwritten signatures through deep learning with resnet-50 and ocr. *International Journal of Innovations in Engineering Research and Technology*, 11:20–25, 3 2024.

[194] Tianyu Liu, Boya Di, and Lingyang Song. Privacy-preserving federated edge learning: Modeling and optimization. *IEEE Communications Letters*, 26:1489–1493, 7 2022.

[195] Akarsh K. Nair, Jayakrushna Sahoo, and Ebin Deni Raj. Privacy preserving federated learning framework for iomt based big data analysis using edge computing. *Computer Standards & Interfaces*, 86:103720, 8 2023.

[196] Wenbo Yang, Hao Wang, Zhi Li, Ziyu Niu, Lei Wu, XiaoChao Wei, Ye Su, and Willy Susilo. Privacy-preserving machine learning in cloud–edge–end collaborative environments. *IEEE Internet of Things Journal*, 12:419–434, 1 2025.

[197] Xiong Li, Jiabei He, Pandi Vijayakumar, Xiaosong Zhang, and Victor Chang. A verifiable privacy-preserving machine learning prediction scheme for edge-enhanced hcpss. *IEEE Transactions on Industrial Informatics*, 18:5494–5503, 8 2022.

[198] Qipeng Xie, Siyang Jiang, Linshan Jiang, Yongzhi Huang, Zhihe Zhao, Salabat Khan, Wangchen Dai, Zhe Liu, and Kaishun Wu. Efficiency optimization techniques in privacy-preserving federated learning with homomorphic encryption: A brief survey. *IEEE Internet of Things Journal*, 11:24569–24580, 7 2024.

[199] Reshma Sheik, Sneha Rao Ganta, and S. Jaya Nirmala. Legal sentence boundary detection using hybrid deep learning and statistical models. *Artificial Intelligence and Law*, 33:519–549, 6 2025.

[200] Reshma Sheik, Gokul T, and S Nirmala. Efficient deep learning-based sentence boundary detection in legal text. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 208–217. Association for Computational Linguistics, 2022.

[201] R. Geetha, T. Thilagam, and T. Padmavathy. Retracted article: Effective offline handwritten text recognition model based on a sequence-to-sequence approach with cnn–rnn networks. *Neural Computing and Applications*, 33:10923–10934, 9 2021.

[202] Tahira Shehzadi, Didier Stricker, and Muhammad Zeshan Afzal. A hybrid approach for document layout analysis in document images. In *International Conference on Document Analysis and Recognition*, pages 21–39. Springer, 2024.

[203] Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. Dit: Self-supervised pre-training for document image transformer. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3530–3539. ACM, 10 2022.

[204] Lukasz Garncarek, Rafal Powalski, Tomasz Stanislawek, Bartosz Topolski, Piotr Halama, and Filip Gralinski. LAMBERT: layout-aware language modeling using BERT for information extraction. *CoRR*, abs/2002.08087, 2020.

[205] Ha-Thanh Nguyen, Minh-Phuong Nguyen, Thi-Hai-Yen Vuong, Minh-Quan Bui, Minh-Chau Nguyen, Tran-Binh Dang, Vu Tran, Le-Minh Nguyen, and Ken Satoh. Transformer-based approaches for legal text processing. *The Review of Socionetwork Strategies*, 16:135–155, 4 2022.

[206] Jiayu Yuan. Efficient techniques for processing medical texts in legal documents using transformer architecture. In *2024 4th International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC)*, pages 990–993. IEEE, 12 2024.

[207] Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikar Appalaraju, and R. Manmatha. Latr: Layout-aware transformer for scene-text vqa. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16527–16537. IEEE, 6 2022.

[208] Sagar Chakraborty, Gaurav Harit, and Saptarshi Ghosh. Transdocanalyser: A framework for offline semi-structured handwritten document analysis in the legal domain, 2023.

[209] Yung-Hsin Chen and Phillip B. Ströbel. Trocr meets language models: An end-to-end post-correction approach. In Harold Mouchère and Anna Zhu, editors, *Document Analysis and Recognition – ICDAR 2024 Workshops*, pages 12–26, Cham, 2024. Springer Nature Switzerland.

[210] Lamia Mosbah, Ikram Moalla, Tarek M. Hamdani, Bilel Neji, Taha Beyrouthy, and Adel M. Alimi. Adocrnet: A deep learning ocr for arabic documents recognition. *IEEE Access*, 12:55620–55631, 2024.

[211] Amirreza Fateh, Reza Tahmasbi Birgani, Mansoor Fateh, and Vahid Abolghasemi. Advancing multilingual handwritten numeral recognition with attention-driven transfer learning. *IEEE Access*, 12:41381–41395, 2024.

[212] Deepak Sinwar, Vijaypal Singh Dhaka, Nitesh Pradhan, and Saumya Pandey. Offline script recognition from handwritten and printed multilingual documents: a survey. *International Journal on Document Analysis and Recognition (IJDAR)*, 24:97–121, 6 2021.

[213] Tayyab Nasir, Muhammad Kamran Malik, and Khurram Shahzad. Mmu-ocr-21: Towards end-to-end urdu text recognition using deep learning. *IEEE Access*, 9:124945–124962, 2021.

[214] Ha-Thanh Nguyen, Manh-Kien Phi, Xuan-Bach Ngo, Vu Tran, Le-Minh Nguyen, and Minh-Phuong Tu. Attentive deep neural networks for legal document retrieval. *Artificial Intelligence and Law*, 32:57–86, 3 2024.

[215] Divya Mohan and Latha Ravindran Nair. Deep learning-based semantic segmentation for legal texts: Unveiling rhetorical roles in legal case documents. In *E3S Web of Conferences*, volume 529, page 04019. EDP Sciences, 2024.

[216] Vijit Malik, Rishabh Sanjay, Shouvik Kumar Guha, Angshuman Hazarika, Shubham Nigam, Arnab Bhattacharya, and Ashutosh Modi. Semantic segmentation of legal documents via rhetorical roles. *arXiv preprint arXiv:2112.01836*, 2021.

[217] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[218] ABBYY. Abbyy software helps streamline a key business process for top london law firm, 2013.

[219] ABBYY. Legal giant norton rose centralises document digitising with abbyy recognition server, 2014.

[220] Tejalal Choudhary, Vipul Mishra, Anurag Goswami, and Jagannathan Sarangapani. A comprehensive survey on model compression and acceleration. *Artificial Intelligence Review*, 53:5113–5155, 10 2020.

[221] Xue Geng, Zhe Wang, Chunyun Chen, Qing Xu, Kaixin Xu, Chao Jin, Manas Gupta, Xulei Yang, Zhenghua Chen, and Mohamed M Sabry Aly. From algorithm to hardware: A survey on efficient and safe deployment of deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[222] Chao Chen, Bohang Jiang, Shengli Liu, Chuanhuang Li, Celimuge Wu, and Rui Yin. Efficient federated learning in resource-constrained edge intelligence networks using model compression. *IEEE Transactions on Vehicular Technology*, 73:2643–2655, 2 2024.

[223] Yiming Chen, Lusine Abrahamyan, Hichem Sahli, and Nikos Deligiannis. Learned parameter compression for efficient and privacy-preserving federated learning. *IEEE Open Journal of the Communications Society*, 5:3503–3516, 2024.

[224] Muhammad Ayat Hidayat, Yugo Nakamura, and Yutaka Arakawa. Privacy-preserving federated learning with resource-adaptive compression for edge devices. *IEEE Internet of Things Journal*, 11:13180–13198, 4 2024.

[225] Bin Jiang, Jianqiang Li, Huihui Wang, and Houbing Song. Privacy-preserving federated learning for industrial edge computing via hybrid differential privacy and adaptive compression. *IEEE Transactions on Industrial Informatics*, 19:1136–1144, 2 2023.

[226] Xi Zhu, Junbo Wang, Wuhui Chen, and Kento Sato. Model compression and privacy preserving framework for federated learning. *Future Generation Computer Systems*, 140:376–389, 3 2023.

[227] Xidi Qu, Qin Hu, and Shengling Wang. Privacy-preserving model training architecture for intelligent edge computing. *Computer Communications*, 162:94–101, 10 2020.

[228] Jiří Martínek, Ladislav Lenc, and Pavel Král. Building an efficient ocr system for historical documents with little training data. *Neural Computing and Applications*, 32:17209–17227, 12 2020.

[229] Christian Clausner, Apostolos Antonacopoulos, and Stefan Pletschacher. Efficient and effective ocr engine training. *International Journal on Document Analysis and Recognition (IJDAR)*, 23:73–88, 3 2020.

[230] Naina Handa, Anil Sharma, and Amardeep Gupta. Framework for prediction and classification of non functional requirements: a novel vision. *Cluster Computing*, 25:1155–1173, 4 2022.

[231] Muhammad Amin Khan, Muhammad Sohail Khan, Inayat Khan, Shafiq Ahmad, and Shamsul Huda. Non functional requirements identification and classification using transfer learning model. *IEEE Access*, 11:74997–75005, 2023.

[232] Sandeep Gupta. Non-functional requirements elicitation for edge computing. *Internet of Things*, 18:100503, 5 2022.

[233] Oshri Naparstek, Ophir Azulai, Daniel Rotman, Yevgeny Burshtein, Peter Staar, and Udi Barzelay. Businet–a light and fast text detection network for business documents. *arXiv preprint arXiv:2207.01220*, 2022.

[234] Sugam Garg, Harichandana SS, and Sumit Kumar. On-device document classification using multimodal features. In *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)*, pages 203–207, 2021.

[235] Clemens Neudecker, Konstantin Baierer, Mike Gerber, Christian Clausner, Apostolos Antonacopoulos, and Stefan Pletschacher. A survey of ocr evaluation tools and

metrics. In *The 6th International Workshop on Historical Document Imaging and Processing*, pages 13–18. ACM, 9 2021.

Appendix

Detailed ABBYY SDK Confidence Score Analysis

1 Experimental Analysis of OCR Confidence Scores Using ABBYY SDK

This section contains the experimental evaluation of confidence values derived from the ABBYY SDK OCR demo, which provides character-level confidence values in an XML output file. The methodology involves analysing the XML data, calculating word-level reliability metrics, comparing the OCR results with the reference data, and visualising the relationship between reliability values and recognition accuracy, with a particular focus on misrecognised words.

1.1 Data Processing and XML Analysis

The experiment begins with the analysis of the XML file generated by the ABBYY SDK (`ocr_result3.xml`), which contains the confidence values for each recognised character.

A Python script extracts the `wordFirst` attribute to identify word boundaries, the recognised character and its confidence value. This data is stored in a CSV file (`ocr_characters.csv`) with three fields: `word_start`, `character` and `confidence`. This step converts the raw XML into a structured format for further analysis.

1.2 Word Reconstruction and Per-Word Confidence Score Generation

Using the file `ocr_characters.csv`, the words are reconstructed by grouping character strings based on the indicator `word_start`, which marks the beginning of each new word. Each character is assigned an individual confidence score provided by the OCR system.

To evaluate the overall reliability of OCR recognition at the word level, six different aggregation methods are applied to the character-level confidence scores within each word. These include the arithmetic mean (`avg_confidence`), the minimum (`min_confidence`), the maximum (`max_confidence`), the median (`median_confidence`), the character width-weighted average (`weighted_confidence`), and the harmonic mean (`harmonic_conf`).

The resulting word-level confidence scores are stored in `words_confidence.csv`, which allows for a comparative analysis of these aggregation strategies to determine which best reflects the actual recognition accuracy. This approach provides a more nuanced assessment of OCR reliability, as it captures both the central trend and the fluctuations in character-level reliability within each word.

1.3 Comparison and Alignment with the Ground Truth

The evaluation process begins with the extraction of words from `words_confidence.csv`, which are then saved in `abby_sdk_ocr.txt`. These OCR-generated words are then compared with a ground truth dataset stored in the `151-2024_Διορθωμένη.docx` file in the `abby_sdk_diff_ground_truth` folder.

The comparison is performed using a differential analysis that systematically identifies discrepancies between the OCR output and the ground truth. This method uses a sequence alignment technique to classify differences into four different operations: `equal` for words that match perfectly, `replace` for words where the OCR output replaces the ground truth data, `delete` for words present in the ground truth data but absent in the OCR output, and `insert` for words present in the OCR output but absent in the ground truth data. The results of this analysis are collected in `abby_sdk_ground_truth_diff.csv`, where the type of operation, the ground truth word and the corresponding OCR word are documented for each comparison.

After analysing the differences, `words_confidence.csv` is compared with `abby_sdk_ground_truth_diff.csv` to generate `word_confidence_with_state.csv`.

This aligned dataset extends the word-level confidence values with a `state` field that classifies each word as correctly recognised (`equal`) or incorrectly recognised (`replace`, `delete` or `insert`).

To enable a thorough evaluation of recognition accuracy and improve alignment, words are extracted together with their confidence values from `word_confidence_with_state.csv` and printed for visual verification. This step allows for a detailed evaluation of the relationship between confidence values and recognition errors and supports the identification of patterns that can provide useful information for future optimisation strategies.

The difference analysis produced the following results, summarised in Table .1 for clarity:

Table .1: Summary of the results of the analysis of differences

Category	Number
Total number of words compared	11,719
Correct words (equal)	11,351
Incorrect words (total errors)	368
- Replaced words (replace)	111
- Deleted words (delete)	3
- Inserted words (insert)	254

These results show a high recognition rate with a low percentage of errors, mainly due to insertion operations, which indicates possible areas for improvement in the performance of the OCR system.

1.4 Visualisation and Analysis of Incorrect Words

To evaluate the predictive power of confidence values in identifying recognition errors, a comprehensive visualisation strategy is used that focuses on words classified according to their correspondence with reference data. The process uses data from the `abby_sdk_diff_ground_truth` folder, which contains the files needed to compare `abby_sdk_ocr.txt` with the reference data.

The results of the difference analysis, stored in `abby_sdk_ground_truth_diff.csv`, provide word-level accuracy assessments. These results are integrated into the `word_confidence_with_state.csv` file, which adds a `state` field to the `words_conf-`

idence.csv file indicating whether each word was recognised correctly or incorrectly, based on the difference analysis.

Initial visualisations are created to examine the distribution of confidence values across all words. The confidence_metrics_by_state.png graph shows scatter plots for each confidence metric—mean, minimum, maximum, median, weighted, and harmonic—with the y-axis representing the confidence value and the x-axis representing the word index. Correctly recognised words are marked in green, while incorrectly recognised words are highlighted in red, thus facilitating a comparative analysis of confidence patterns.

The following analysis focuses exclusively on incorrectly recognised words in order to evaluate the effectiveness of confidence values as error indicators. Specific graphs are created for each confidence metric: wrong_words_avg_confidence.png and wrong_words_median_confidence.png show the average and median values of confidence values for non-consecutive incorrect words in two segments of the dataset, comprising 59 and 60 words respectively.

Additional graphs are created, including wrong_words_min_confidence.png, for minimum, maximum, weighted, and harmonic confidence values (which are not all shown here, however). These scatter plots show confidence values between 0 and 100%, with horizontal lines at 80% (high quality threshold) and 30% (low quality threshold) as reference points for quality assessment.

1.5 Conclusion: Confidence Score Analysis Results

Confidence Score Performance Across All Metrics

The scatter plots illustrate the distribution of reliability values for six metrics (y-axis: 0–100) in relation to the word index (x-axis) with colour coding: green for correctly recognised words (11,303 cases), red for incorrect/misplaced words (298 cases), grey for unknown quality (115 cases), blue for high-quality annotations (80 cases) and yellow for low-quality annotations (30 cases).

The avg_confidence and weighted_confidence diagrams show green dots grouped between 60 and 100, with red dots scattered between 0 and 100. The min_confidence diagram shows a large scattering of green and red points, making it difficult to distinguish between them, while max_confidence concentrates both points in higher intervals, thus limiting the distinction. The median_confidence and harmonic_confidence

diagrams show intermediate patterns, with `median_confidence` providing a structured separation of green (above 60) and red (below 60) points.

Overall Performance Analysis

The correlations between confidence value and OCR accuracy range from 0.044 (`min_confidence`) to 0.180 (`median_confidence`), influenced by Ground Truth limitations. Manual annotation errors occur when words correctly recognised in the manually generated ground truth are incorrectly marked as incorrect, artificially reducing the apparent correlation. Inconsistencies in distances occur when words appear as separate tokens in the OCR output but are connected in the ground truth and are marked as incorrect, despite reflecting formatting issues rather than recognition issues.

The `median_confidence` metric proves to be the most robust, as it compensates for the effects of outliers and allows for a clearer separation between correct and incorrect words, as demonstrated by the structured distribution of the scatter plot despite these data imperfections.

Conclusion: Error Word Analysis and Confidence Score

Analysis of the reliability values of incorrectly recognised words reveals significant limitations in the assessment of the reliability of the ABBYY SDK. Many false words have high reliability values across all parameters, indicating a systematic overconfidence in false predictions. For example, the word ’πόδατου’ (an OCR error) received high reliability values in all parameters:

- `avg_confidence`: 91.7
- `min_confidence`: 69.0
- `max_confidence`: 100.0
- `median_confidence`: 100.0
- `weighted_confidence`: 91.3
- `harmonic_confidence`: 90.1

However, the confidence system works better for errors in handwritten text, where `min_confidence` provides more reliable error indicators. For example, the incorrectly recognised handwritten sequence ’1S1./2024’ has lower confidence values:

- `avg_confidence`: 12
- `min_confidence`: 100
- `max_confidence`: 52.0
- `median_confidence`: 56.3
- `weighted_confidence`: 28.4
- `harmonic_confidence`: 77

Despite its effectiveness with handwritten text, `min_confidence` suffers from high false positive rates and often reports correctly recognised words as suspicious.

The inconsistent performance, with some errors retaining high confidence while others receive appropriately low values, shows that confidence metrics cannot serve as standalone error detection tools. The analysis suggests that confidence values are more useful as complementary metrics within comprehensive error detection pipelines, particularly for identifying potential errors in handwritten text areas.