Information Processing and Management 64 (2027) 105047

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

Assessing and comparing document OCR systems in the era of Large Language Models

Valerio Caravani [a,b,*], Riccardo De Cesaris [a], Paolo Merialdo [a]

[a] Department of Engineering, Roma Tre University, Via Vito Volterra 62, 00146 Rome, Italy
[b] myBiros.com, Via Antonio Schivardi 60, 00144, Rome, Italy

ARTICLE INFO

Keywords:
Optical character recognition (OCR)
Multimodal Large Language Models (LLMs)
Assessment
Evaluation metrics
Industrial document processing
Open-source and commercial systems

ABSTRACT

Optical Character Recognition (OCR) is a fundamental component of document analysis pipelines, enabling downstream tasks such as information extraction, retrieval, and serving as a key enabler of automation. The recent emergence of multimodal Large Language Models (LLMs) has introduced alternative approaches that integrate implicit OCR capabilities within unified vision–language architectures. Despite their growing prominence, a systematic comparison between traditional OCR systems and multimodal LLMs is still lacking. This paper presents a comprehensive benchmarking study encompassing 16 OCR and multimodal systems, grouped into open-source OCR engines, commercial OCR services, commercial multimodal LLMs, and open-source multimodal LLMs. The evaluation spans four publicly available datasets representing printed, scanned, handwritten, English and Chinese dense-text multicolumn documents. Performance is assessed using character- and word-level accuracy, normalized edit distance, and flexible character accuracy, complemented by latency and cost analyses to provide a holistic view of each system's operational suitability. The results show that no single paradigm excels universally: in our settings, multimodal LLMs achieve stronger performance on unstructured and handwritten inputs, while specialized OCR pipelines remain more reliable for structured and printed layouts. Lightweight domain adaptation through fine-tuning proves beneficial for open-source OCR models, whereas general-purpose LLMs offer competitive accuracy at higher computational costs. All assessment tools and configurations are publicly released to promote reproducibility and facilitate future OCR and LLM research within the document analysis community.

1. Introduction

OCR is a core component of document processing, which converts scanned or photographed documents into machine-readable text. It supports a wide range of industrial applications, including large-scale digital archiving, information retrieval, and automation of business processes (Liu et al., 2019; Singh et al., 2012; Song, 2026). Beyond standalone transcription, OCR also provides the foundation for advanced document understanding workflows, allowing downstream tasks such as named-entity recognition, key–value extraction, table parsing, and document-level question answering (Gbada et al., 2025; Liao et al., 2023; Rashid et al., 2017; Yan et al., 2025). Ensuring high accuracy at the OCR stage is critical, as transcription errors can propagate and compromise the reliability of subsequent tasks (Ghosh et al., 2016; Lam-Adesina & Jones, 2006; van Strien et al., 2020).

* Corresponding author at: Department of Engineering, Roma Tre University, Via Vito Volterra 62, 00146 Rome, Italy.
E-mail addresses: valerio.caravani@uniroma3.it (V. Caravani), riccardo.decesaris@uniroma3.it (R. De Cesaris), paolo.merialdo@uniroma3.it (P. Merialdo).

https://doi.org/10.1016/j.ipm.2026.105047
Received 21 October 2025; Received in revised form 17 June 2026; Accepted 4 July 2026
Available online 23 July 2026

In industrial document processing, both machine-printed text recognition (OCR) and handwritten text recognition (HTR) are relevant paradigms, addressing complementary document types and posing distinct technical challenges. This study adopts the term OCR broadly to encompass both, consistently with common usage in the document analysis community.

Substantial progress has been made in OCR over the past few years (Gbada et al., 2025; Wang et al., 2023). Deep learning-based OCR systems have evolved into highly optimized and widely adopted solutions, while multimodal LLMs have recently emerged as an alternative approach, embedding implicit OCR capabilities within end-to-end architectures and thereby simplifying both pipeline design and generalization (Yin et al., 2024). This variety of solutions introduces different trade-offs in terms of accuracy, computational and monetary costs, and processing time.

Beyond transcription accuracy, the adoption of OCR systems in industrial settings is shaped by a broader set of operational and strategic considerations. Processing latency directly affects the throughput of document pipelines and the feasibility of real-time applications. Monetary cost determines the scalability of a solution, particularly when processing large document volumes under budget constraints. Privacy and compliance requirements, which are increasingly stringent under regulations such as GDPR, may preclude the use of cloud-based services that transmit sensitive document content to third-party APIs, making on-premise open-source solutions preferable in regulated industries such as healthcare, finance, and legal services. The emergence of multimodal LLMs introduces new trade-offs along all these dimensions: while they offer compelling accuracy and generalization, their reliance on external APIs, higher per-page costs, and non-deterministic outputs raise practical concerns for production deployment. A rigorous comparative evaluation must therefore go beyond accuracy and systematically address these operational dimensions.

With the growing prominence of LLMs, this work seeks to assess how general-purpose multimodal models compare with specialized OCR solutions through a systematic comparative evaluation encompassing 16 systems, organized into four categories: open-source OCR engines, commercial cloud-based OCR services, commercial multimodal LLMs, and open-source multimodal LLMs. The evaluation covers 4 publicly available datasets representing diverse real-world scenarios: printed receipts, scanned forms, handwritten notes, and dense-text multi-column documents in English and Chinese. Performance assessment is evaluated along three dimensions: transcription accuracy, processing latency, and operational cost, providing a holistic view of each system's suitability for industrial deployment.

OCR benchmarking remains an active research area. Recent efforts such as OCRBench V2 (Fu et al., 2024) and OmniDocBench (Ouyang et al., 2025) represent significant advances, but address complementary and distinct problems. OCRBench V2 focuses exclusively on evaluating multimodal LLMs across text-oriented visual understanding tasks, without including traditional or commercial OCR engines and without considering operational dimensions such as processing latency or monetary cost. OmniDocBench targets document parsing systems, evaluating the conversion of PDF documents into structured representations, rather than raw text transcription from scanned or photographed images, and similarly does not address operational efficiency. Other recent benchmarks, such as CC-OCR (Yang et al., 2024) and WildDoc (Wang et al., 2025), further extend MLLM evaluation to multi-scene and real-world capture conditions, but remain likewise focused on model accuracy without incorporating traditional OCR engines or operational dimensions. Consequently, a unified benchmarking framework that jointly evaluates traditional OCR engines, commercial cloud services, and multimodal LLMs, across both accuracy and operational dimensions such as latency and cost, and that targets real-world document transcription scenarios, remains absent from the literature.

The main contributions of this work are:

- **Comprehensive Evaluation.** We conduct a systematic comparison of text recognition performance across open-source and commercial OCR engines and recent multimodal LLMs, both proprietary and open-source, analyzing accuracy, processing latency, and monetary cost on publicly available datasets representative of real-world document types. Unlike existing benchmarks such as OCRBench V2 (Fu et al., 2024) and OmniDocBench (Ouyang et al., 2025), which focuses exclusively on multimodal LLMs without addressing operational efficiency, our framework introduces cost and latency as first-class evaluation dimensions alongside accuracy, enabling a holistic assessment of industrial applicability.
- **Unified Benchmarking Framework.** We introduce a consistent and reproducible evaluation pipeline that harmonizes heterogeneous system outputs and applies four OCR metrics, ensuring a fair comparison across 16 systems. The evaluation is conducted on four publicly available datasets, selected to cover a broad spectrum of real-world document types, including printed receipts, noisy scanned forms, handwritten text, and dense-text multi-column documents in English and Chinese. This combination ensures that conclusions are grounded in diverse visual, structural, and linguistic conditions rather than a narrow set of scenarios.
- **Empirical and Analytical Insights.** We identify the operational strengths and limitations of both traditional OCR pipelines and LLM-based approaches, highlighting the trade-offs that shape their industrial applicability.

To ensure reproducibility and facilitate future benchmarking and model extensions, the experimental evaluation code will be made available through the project repository.[1]

The paper is organized as follows. Section 2 reviews related work. Section 3 provides the technical background, and Section 4 describes the evaluated system categories. Section 5 introduces the methodology and datasets, and Section 6 presents the assessment results, and Section 7 summarizes the conclusions of the assessment.

[1] https://github.com/valcar-myb/assessing-and-compare-document-ocr-systems-in-the-era-of-llms

2. Related work

This section reviews related work in OCR systems evaluation, with a focus on studies that have compared open-source, commercial, and more recently, multimodal LLMs.

2.1. Open-source and commercial OCR benchmarks

One of the earliest systematic efforts in OCR evaluation was the *Annual Test of OCR Accuracy* series, conducted by Rice et al. (1993), which established the methodological foundations of modern benchmarking practices. It remains influential for introducing standard metrics such as Character and Word Accuracy (Rice et al., 1993), incorporating time measurements, and covering diverse document types and resolutions. Although historically important, the study is now dated, relying on a non-public dataset and legacy systems that predate modern cloud-based and deep learning-based OCR approaches.

Several early comparative studies (Gabasio, 2013; Vijayarani & Sakila, 2015; Vithlani & Kumbharana, 2015) evaluated both open-source and commercial OCR tools, mostly web-based or desktop solutions. Tafti et al. (2016) later conducted one of the first large-scale benchmarks on over 1200 printed documents, confirming the superior accuracy of commercial engines over open-source counterparts. Reul et al. (2018) subsequently conducted a systematic comparison between open-source engines and a leading commercial OCR system, showing that modern open solutions can reach or surpass commercial performance when properly trained. Hegghammer (2022) compared Tesseract with commercial cloud services (Amazon Textract, Google Document AI) using noisy inputs, showing that commercial systems were generally more noise-tolerant. Recently a domain-specific benchmark was presented by Batra et al. (2024), who evaluated three open-source OCR engines such as Tesseract (Smith, 2007) and Doctr (Mindee Team, 2021) on the task of medical report digitization. The study focused on the impact of image pre-processing techniques on recognition accuracy, but was limited to a single domain and did not include commercial systems.

2.2. Benchmarking on LLMs as OCR engines

With the advent of vision-enabled LLMs, research has increasingly explored their potential as OCR engines. Shi et al. (2023) conducted one of the earliest evaluations of GPT-4V on document-related tasks such as OCR and table extraction, preceding more extensive benchmark initiatives. Fu et al. (2024), Y. Liu et al. (2024) introduced *OCRBench* and *OCRBench V2*, assessing dozens of multimodal LLMs across up to 31 datasets and 23 OCR-related tasks, ranging from scene text to handwritten and multilingual recognition. Yang et al. (2024) proposed *CC-OCR*, a comprehensive benchmark spanning 39 sub-tasks, while Wang et al. (2025) developed *WildDoc* to evaluate robustness under real-world capture conditions such as illumination and distortion.

More recently, Ouyang et al. (2025) introduced *OmniDocBench*, a large-scale benchmark for comprehensive evaluation of document parsing systems. It provides fine-grained annotations across nine PDF types, including academic papers, textbooks, handwritten notes, and newspapers.

Despite the growing coverage offered by recent benchmarks, current studies address complementary but distinct problems. OCRBench V2 (Fu et al., 2024) focuses exclusively on evaluating multimodal LLMs across text-oriented visual understanding tasks, without including traditional or commercial OCR engines and without considering operational dimensions such as processing latency or monetary cost. OmniDocBench (Ouyang et al., 2025) targets document parsing systems, evaluating the conversion of PDF documents into structured representations, rather than raw text transcription from scanned or photographed images, and similarly does not address operational efficiency. Consequently, a unified benchmarking framework that jointly evaluates traditional OCR engines, commercial cloud services, and multimodal LLMs across both accuracy and operational dimensions such as latency and cost, targeting real-world document transcription scenarios, remains absent from the literature.

To summarize, existing benchmarks consistently focus on a subset of multimodal LLMs, excluding traditional open-source OCR engines and commercial cloud-based OCR services that remain widely deployed in industrial settings. Furthermore, none of the existing frameworks incorporates latency or monetary cost as evaluation dimensions, limiting their practical relevance for deployment decisions. Our benchmark addresses these gaps by providing a unified evaluation framework that spans all four system categories (open-source OCR engines, commercial OCR services, commercial multimodal LLMs, and open-source multimodal LLMs) and complements accuracy-based metrics with a systematic analysis of processing time and operational cost across diverse document types in English and Chinese.

3. Background

This section introduces the two main system families examined in this study, traditional OCR engines and multimodal large language models (LLMs), and discusses how their architectures, processing pipelines, and learning paradigms affect text recognition performance and generalization.

3.1. Traditional OCR systems

Traditional OCR systems typically adopt a modular architecture composed of two main stages (Zhu et al., 2016): *Text detection* and *Text recognition*. The first stage aims at identifying regions of interest within the document by localizing bounding boxes containing textual content. The second stage transcribes the detected regions into character sequences.

Early modular architectures. Early OCR engines were heavily dependent on explicit pre- and post-processing to deal with noise, skew, and layout variability (Smith, 2007; Ye & Doermann, 2015). Common preprocessing techniques included binarization and deskewing (Otsu et al., 1975; Smith, 2007), while post-processing relied on dictionaries or language models for error correction (Kukich, 1992; Mei et al., 2018; Reynaert, 2008). Text detection was typically implemented using connected-component analysis or thresholding (Neumann & Matas, 2012; Ye & Doermann, 2015), which were computationally efficient but often unreliable for complex layouts or handwritten text (Jo et al., 2020). For text recognition, early methods employed handcrafted features and statistical sequence models such as Hidden Markov Models (Agazzi & Kuo, 1993). These systems achieved reasonable accuracy on clean, printed documents but performed poorly under varying document quality and layout conditions (Zhu et al., 2016).

Deep learning–based architectures. The advent of deep learning fundamentally evolved both detection and recognition stages. Convolutional neural networks (CNNs) substantially improved text localization, with detectors such as DBNet (Liao et al., 2020) achieving strong performance even on irregular or curved text (Zhang et al., 2020). Recognition models evolved from statistical approaches to deep sequence models trained with Connectionist Temporal Classification (CTC) loss (Graves et al., 2006). The CRNN architecture (Shi et al., 2017), combining convolutional encoders with recurrent LSTM layers, became a cornerstone in alignment-free transcription. More recently, Transformer-based architectures (Li et al., 2023) have achieved state-of-the-art results across fonts, languages, and complex layouts by leveraging self-attention and large-scale *pretraining*. In these systems, robustness once handled by explicit preprocessing is instead learned through extensive data augmentation during training (Li et al., 2023; Shi et al., 2017).

Model adaptation and fine-tuning. While pretrained OCR models deliver strong performance on general-purpose documents, domain adaptation remains essential to achieve high accuracy in specialized settings. Pretraining typically relies on large and diverse synthetic datasets, which provide strong generalization capabilities, whereas *fine-tuning* on smaller, domain-specific corpora allows models to adapt to particular fonts, languages, noise patterns, or layouts (Veit et al., 2016).

3.2. Multimodal LLMs

The rapid progress of vision–language foundation models has profoundly transformed the landscape of document understanding (Yin et al., 2024). Recent multimodal LLMs integrate vision encoders and text decoders within unified Transformer-based architectures, enabling direct visual–language generation without explicit OCR stages (Chen et al., 2023; Laurençon et al., 2024; Zhang et al., 2023). Their main strength lies in generalization. Trained on large-scale image–text corpora, these models naturally extend to downstream document tasks such as key–value extraction, table parsing, and question answering (Q. Team, 2024), and can also perform text transcription when prompted accordingly.

Deployment and accessibility of multimodal models. The computational and infrastructural demands of hosting large-scale architectures make self-deployment challenging for most organizations. Consequently, the initial adoption of general-purpose multimodal LLMs largely occurred through commercial API-based systems such as *GPT-4V* (Achiam et al., 2023) and *Gemini 1.5* (Team et al., 2024), which first made large-scale vision–language capabilities broadly available. In parallel, the open research community has increasingly focused on developing more lightweight multimodal LLMs suitable for local integration. Models such as *Qwen2.5-VL* and *Gemma 3* (G. Team, 2025; Q. Team, 2024) exemplify this trend, offering competitive vision–language capabilities under permissive licenses that enable cost-effective on-premise use.

Current limitations and specialization. Nevertheless, important trade-offs with established OCR systems remain. Unlike traditional OCR engines, multimodal LLMs often lack structured outputs such as bounding boxes, layout metadata, and confidence scores, which are essential for reliable integration into production pipelines. While general-purpose models such as *Qwen2.5-VL* (Q. Team, 2024) and *Gemini 2.0* (Google, 2025) have begun to incorporate capabilities for structured document understanding, these remain limited and non-standardized. For these reasons, specialized multimodal LLMs tailored for document processing are emerging (Mistral AI, 2024), aiming to bridge this gap by providing the structure and reliability required for industrial-grade applications.

4. Evaluated systems

The families of systems outlined in Section 3 are reflected in the systems evaluated in this study. These include both open-source frameworks and commercial services designed for document transcription. In total, 16 systems were evaluated, organized into four categories, as shown in Fig. 1: *(i)* open-source OCR engines, *(ii)* commercial cloud-based OCR services, *(iii)* commercial multimodal LLMs, and *(iv)* open-source multimodal LLMs. These systems were selected for their widespread adoption, accessibility, and suitability for a broad range of document processing tasks, spanning from established open-source solutions to production-grade services.

4.1. Open-source OCR engines

Open-source OCR engines are pivotal in academic research and industrial applications due to their transparency, permissive licensing, and adaptable deployment options. These features make them particularly suitable for privacy-sensitive environments. However, achieving optimal performance in production often demands domain-specific expertise and careful optimization. In this study, we evaluated three widely adopted open-source OCR frameworks, each representing a distinct architectural approach as summarized in Table 1. The evaluated frameworks are:

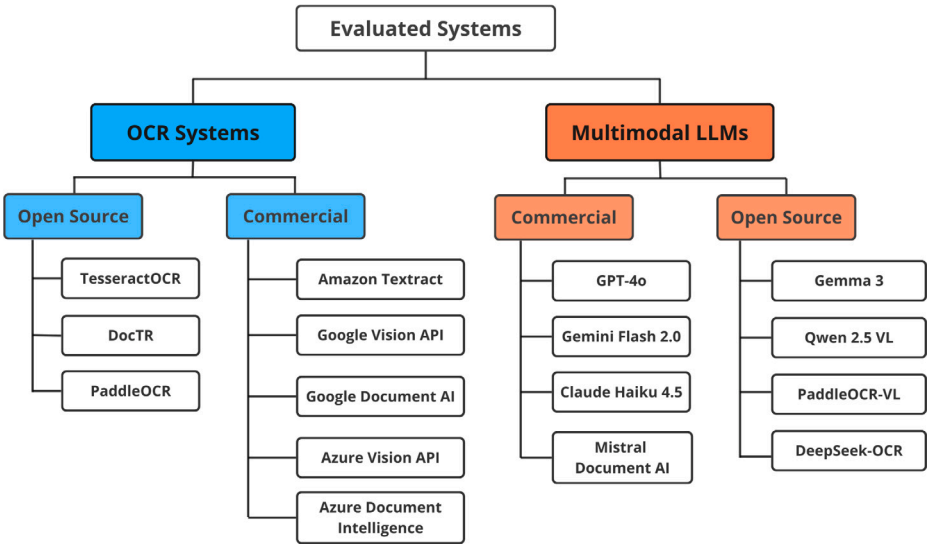

Fig. 1. Evaluated OCR and multimodal LLM systems, grouped by open-source and commercial categories.

Table 1
Architectural overview and GPU support of the evaluated open-source OCR engines.

OCR engine	Version	Detection stage	Recognition stage	GPU support
Tesseract	5.0	Default (PSM=3)	LSTM-CTC (eng)	No
DocTR	0.8.1	DBNet (ResNet-50)	CRNN (VGG16-BN)	Yes
PaddleOCR	2.3	MobileNetV3	SVTR-LCNet	Yes

- **Tesseract** (Smith, 2007), a widely used engine maintained by Google. Text detection is performed implicitly through page layout and component analysis, while recognition relies on an LSTM-based model trained with CTC loss.
- **DocTR** (Mindee Team, 2021), a modular deep learning library offering pretrained models for both text detection and recognition. In our setup, detection was performed using DBNet with a ResNet-50 (He et al., 2016) backbone, paired with a CRNN recognizer based on VGG16-BN (Simonyan & Zisserman, 2015).
- **PaddleOCR** (Cui, Sun, Lin, et al., 2025), an industrial-grade system built on the PaddlePaddle framework (Ma et al., 2019). In our experiments, we adopted the lightweight PP-OCRv3 configuration, which combines a MobileNetV3-based detector with an SVTR-LCNet recognizer.

DocTR and PaddleOCR both support GPU acceleration and were evaluated on CPU and GPU, while Tesseract was tested only on CPU.

4.2. Commercial OCR service

Commercial OCR services, offered as fully managed cloud platforms by leading providers, deliver scalability, reliability, and minimal setup effort. For this study, we selected the OCR solutions provided by *Microsoft Azure*, *Google Cloud Platform (GCP)*, and *Amazon Web Services (AWS)*, given their widespread adoption and comprehensive support for both natural images and document-centric inputs.

- *Microsoft Azure* provides OCR through two complementary services, both evaluated in this study. **Azure AI Vision**[2] (version 4.0) is optimized for general-purpose image scenarios and exposes a synchronous API suitable for embedding OCR in user-facing applications. **Azure Document Intelligence**[3] is tailored for scanned and digital documents and leverages an asynchronous API designed for large-scale intelligent document processing.
- *GCP*, similarly, offers two distinct OCR solutions. **Cloud Vision OCR**[4] is optimized for text extraction from diverse image inputs, including photographs and natural scenes. **Document AI OCR**[5] targets document-specific use cases and provides enhanced parsing capabilities for structured and unstructured inputs. Both services were tested to capture their respective strengths.

[2] https://learn.microsoft.com/en-us/azure/ai-services/computer-vision/overview-ocr
[3] https://learn.microsoft.com/en-us/azure/ai-services/document-intelligence/overview
[4] https://cloud.google.com/vision/docs/ocr
[5] https://cloud.google.com/document-ai/docs/overview

Table 2
Commercial OCR services evaluated, showing provider, target use case, and model/version control options.

Provider	Service	Target use case	Model/version control
Microsoft Azure	AI Vision (v4.0)	General-purpose images	Yes (version selectable)
Microsoft Azure	Document Intelligence	Document	Yes (model selectable)
Google Cloud	Vision OCR	General-purpose images	Yes (model selectable)
Google Cloud	Document AI OCR	Document	Yes (model selectable)
AWS	Textract	Document	No (latest)

- *AWS* provides OCR capabilities through **Textract**,[6] a service specialized in document analysis. Textract supports tasks such as text recognition, key–value pair extraction, and table detection, and includes asynchronous processing features designed for large-scale enterprise workloads.

A practical difference worth noting is that Azure and Google allow users to specify the model or version used for transcription, whereas AWS Textract abstracts versioning details, potentially affecting reproducibility and long-term stability in production deployments. The main characteristics of the evaluated Commercial OCR services are summarized in Table 2.

4.3. Open-source multimodal LLMs

We evaluated two open-source multimodal LLMs Table 3, selected for their efficient, modern architectures that enable in-house deployment with full data control and moderate computational cost. Both models, accessible through Hugging Face repositories, accept text and images as input and can be prompted for OCR-like document processing tasks.

- **Gemma 3** (G. Team, 2025), developed by Google DeepMind, was evaluated in its 4B instruction-tuned variant.[7] It integrates a SigLIP (Zhai et al., 2023) vision encoder that processes images at a base resolution of 896 × 896 pixels per crop. To handle inputs with varying aspect ratios or high resolutions (such as full A4 document pages), Gemma 3 employs a Pan&Scan mechanism, which adaptively partitions the input image into multiple overlapping crops, resizes each crop independently to 896 × 896 pixels, and encodes them separately. This preserves local text resolution across the full page, at the cost of increased computational overhead proportional to the number of crops generated. The multimodal variant employed here supports a context window of up to 32 K tokens and generates text-only outputs, without support for structured OCR-style features such as bounding boxes or layout metadata.
- **Qwen2.5-VL** (Q. Team, 2024), developed by the Alibaba Qwen team, was evaluated in its 3B instruction-tuned release.[8] It employs a ViT-L (Dosovitskiy, 2020) multimodal encoder trained with a CLIP-style (Radford et al., 2021) objective, typically processing images at resolutions between 448 and 896 pixels. The model supports context lengths of up to 32 K tokens and is capable of producing structured OCR-style outputs, including text transcriptions with bounding-box alignment.
- **PaddleOCR-VL** (Cui, Sun, Liang, et al., 2025) developed by Baidu, was evaluated in its 0.9B variant.[9] It integrates a NaViT-style dynamic resolution visual encoder with the ERNIE-4.5-0.3B language model, enabling accurate recognition of complex document elements including text, tables, formulas, and charts. The model supports 109 languages and is specifically designed for document parsing tasks, combining high recognition accuracy with fast inference speeds and minimal resource consumption, making it well-suited for practical deployment in resource-constrained environments.
- **DeepSeek-OCR** (Wei et al., 2025) developed by DeepSeek, was evaluated in its base variant.[10] It consists of two components: a visual encoder, named DeepEncoder, designed to maintain low token activation under high-resolution inputs while achieving high vision-token compression ratios, built on a SAM/CLIP-based vision backbone, and a DeepSeek3B-MoE-A570M language model as the decoder. The model is specifically optimized for efficient document OCR.

These open models are publicly available and licensed for commercial use under custom agreements, which allow redistribution and deployment under specific conditions.

4.4. Commercial multimodal LLMs

Commercial multimodal LLMs integrate vision and language understanding to perform tasks on sources such as documents and images, and are typically made accessible via conversational interfaces and APIs. Multimodal LLMs can be guided by prompting to produce text transcriptions from images. Representative solutions from leading Generative AI providers were selected and are accessed through APIs; these are summarized in Table 4:

[6] https://docs.aws.amazon.com/textract/latest/dg/what-is.html
[7] https://huggingface.co/google/gemma-3-4b-it
[8] https://huggingface.co/Qwen/Qwen2.5-VL-3B-Instruct
[9] https://huggingface.co/PaddlePaddle/PaddleOCR-VL
[10] https://huggingface.co/deepseek-ai/DeepSeek-OCR

Table 3
Architectural overview and OCR capabilities of the evaluated open-source multimodal LLMs.

Model	Provider	Parameters	Vision encoder	OCR capabilities
Gemma 3	Google	4B	SigLIP ViT	Plain text
Qwen2.5-VL	Alibaba	3B	ViT-L	Plain text + bounding boxes
PaddleOCR-VL	Baidu	0.9B	NaViT	JSON/Markdown metadata
DeepSeek-OCR	DeepSeek	3.3B	DeepEncoder	JSON/Markdown metadata

Table 4
Commercial multimodal LLMs evaluated, showing provider, model version, and OCR-related output capabilities.

Model	Provider	Version	OCR capabilities
GPT-4o	OpenAI	2024–08–06	Plain text
Gemini 2.0 Flash	Google	2024–12	Plain Text + bounding boxes
Claude 4.5 Haiku	Anthropic	2025–10–01	Plain text
Mistral Document AI	Mistral AI	mistral-ocr-2505-completion	JSON/Markdown metadata

- **GPT-4o** (OpenAI): multimodal general-purpose model trained end-to-end across text, vision, and audio, capable of handling inputs in text, audio, image, and video and producing multimodal outputs (Hurst et al., 2024).
- **Gemini 2.0 Flash** (Google): multimodal general-purpose model from the Gemini family, optimized for low latency and extended context (up to 1M tokens). It supports text, image, and audio inputs and is equipped with structured output and object-detection capabilities (Google, 2025).
- **Claude Haiku 4.5** (Anthropic): the fastest and most cost-efficient model in Anthropic's Claude 4.5 family, supporting text and image inputs with text-only outputs. It is a hybrid-reasoning model designed for low-latency and high-throughputworkloads. Anthropic (2025).
- **Mistral Document AI** (Mistral AI): commercial system for document understanding that leverages the company's proprietary language models (Mistral AI, 2024).

5. Assessment method

This section presents the evaluation methodology employed in the assessment. The methodology includes the definition of the recognition task, the selection of datasets, the inference setup, the evaluation metrics, and the fine-tuning procedure applied to an open-source OCR baseline.

5.1. Task definition

We evaluated all systems on the task of full-page text recognition, i.e., transcribing all visible textual content in a document image. Since the outputs of the evaluated systems are heterogeneous, they were converted into a plain text representation containing the predicted character sequence. Systems were assessed directly on the original inputs, without any image pre-processing or output post-processing. For the evaluation, the transcriptions were normalized to a common character set, as detailed in Section 5.4. The procedures used to generate and harmonize the outputs for all systems are documented and available in the project repository.

All systems were evaluated at the single-page level, with each document image submitted as an independent input. This design choice reflects standard practice in OCR pipelines, where multi-page documents are typically processed as sequences of individual pages. Under this assumption, per-page latency generalize directly to multi-page documents, with total processing time scaling linearly with the number of pages.

5.2. Datasets

We selected four publicly available datasets that cover a broad spectrum of real-world document processing scenarios: SROIE (Huang et al., 2019) for printed receipts; IAM (Marti & Bunke, 2002) representing the Handwritten Text Recognition (HTR) setting; FUNSD (Jaume et al., 2019) for noisy scanned forms; and FOX (Liu et al., 2024) for dense-text, multi-column documents in English and Chinese. Together, these datasets provide a comprehensive testbed spanning diverse visual, structural, and linguistic conditions (see Fig. 2).

Each dataset provides ground-truth transcriptions at the line level, which we aggregated into full-page text files to enable comparison with the model predictions. All experiments were conducted using the official training and test splits released by the respective dataset authors. Both IAM and SROIE required minor preprocessing of either the input images or the ground-truth labels, as detailed below. A summary of the datasets used in our assessment is provided in Table 5.

- *SROIE* – The Scanned Receipt OCR and Information Extraction dataset includes 1000 scanned receipts collected for the ICDAR 2019 Robust Reading Challenge. It features low-quality prints, non-standard layouts, and frequent scanning artefacts. Ground-truth annotations are provided in plain-text format at line level. Inconsistencies in case sensitivity were resolved by normalizing all transcriptions to lowercase.

(a) Printed receipts (SROIE) (b) Cursive handwriting (IAM) (c) Noisy scanned forms (FUNSD) (d) Dense-text document pages in English and Chinese (FOX)

Fig. 2. Examples from the evaluation datasets.

Table 5
Summary of the OCR datasets evaluated, with document type, use case, year, and partitions size.

Name	Document type	Year	Size (Train/Test)
SROIE (Huang et al., 2019)	Printed receipts	2019	639/347
IAM (Marti & Bunke, 2002)	Handwritten text	2002	1307/232
FUNSD (Jaume et al., 2019)	Noisy scanned forms	2019	149/50
FOX (Liu et al., 2024)	Dense text (English, Chinese)	2024	—/212

Table 6
Commercial OCR services deployment regions.

Service	Region
AWS Textract OCR API	eu-central-1 (Frankfurt)
Azure (Vision and Document Intelligence)	West Europe (Netherlands)
Google Cloud (Vision and Document AI)	Global

- *FUNSD* – The Form Understanding in Noisy Scanned Documents dataset contains 199 annotated forms extracted from the RVL-CDIP corpus. It is challenging due to low resolution, noise, and diverse layout structures. For the purpose of OCR evaluation, we used only the word-level transcriptions and bounding boxes to generate full-page text files for each document.
- *IAM* – The IAM Handwriting Database consists of 1539 handwritten pages produced by 657 writers. It includes over 115,000 word instances, split into train, validation, and test sets with no writer overlap. Each page contains a printed prompt followed by the handwritten text. Using the official annotations, we cropped out the printed region and retained only the handwritten portion, from which full-page transcription files were generated for evaluation.
- *FOX* – The FOX dataset (Liu et al., 2024) is a dense-text benchmark designed for fine-grained document understanding. It comprises 112 English pages and 100 Chinese pages, featuring single-column, double-column, and mixed-column layouts, with more than 1000 words per page. Ground truth in FOX is available as page-level text serialized in natural reading order. The absence of word/line bounding boxes prevents supervised fine-tuning of traditional OCR pipelines, which makes FOX suitable only for evaluation in our assessment.

5.3. Experimental setup

Experiments were conducted on a dedicated AWS cloud instance (g5.xlarge) equipped with an NVIDIA A10 GPU (24 GB VRAM), 4 vCPUs, and 16 GB RAM, running Ubuntu 22.04 and CUDA 12.2. All API-based services were accessed through requests issued from the AWS Frankfurt region instance, while open-source OCR engines and LLMs were run locally on the same instance.

5.3.1. Commercial OCR services

All experiments were conducted using standard pay-as-you-go cloud accounts at the default service tier. Each service was accessed through its official Python SDK, and one request was submitted per document image. Where available, synchronous APIs were used, while *Google Document AI* and *Azure Document Intelligence* were accessed through their asynchronous endpoints. Table 6 reports the endpoint region used for each service.

Table 7
Prompting strategies adopted for each system category.

Systems	Prompt
General-purpose LLMs (simple prompt)	‘‘Extract all visible text from the document image. Return plain text.’’
General-purpose LLMs (Chain-of-Thought prompt)	‘‘You are an advanced OCR system. Your task is to extract all visible text from the provided document image as accurately as possible. Work through these steps internally before producing your final output. Step 1 - Analyze layout and text areas. Step 2 - Identify text regions: scan the image systematically (top to bottom, left to right), noting overlapping elements, rotated text, or low-contrast areas. Step 3 - Transcribe each region, reasoning through ambiguous characters based on context (e.g., ‘0’ vs ‘O’, ‘1’ vs ‘l’ vs ‘I’). Step 4 - Return only the extracted text.’’
PaddleOCR-VL	‘‘OCR’’
DeepSeek-OCR	‘‘<image> \n Free OCR.’’
Mistral Document AI	No configurable prompt interface.

5.3.2. Commercial LLMs

All commercial multimodal LLMs were accessed through their official APIs and corresponding Python libraries, using the model releases listed in Table 4. We evaluated these general-purpose models under the prompting conditions described in Section 5.3.4, while keeping generation settings as consistent as possible across providers. Where configurable, the temperature parameter was set to 0; otherwise, provider defaults were applied. Each document image was submitted as a single request through the text–image completion API, and models were required to return plain-text transcriptions.

5.3.3. Open-source LLMs

All open-source multimodal LLMs were executed locally using the `vLLM` inference engine (Kwon et al., 2023) (v0.4.2) within Docker containers based on `nvidia/cuda:12.2`. `vLLM` was adopted as a widely used open-source framework that enables efficient inference and exposes an OpenAI-compatible API. It also provides native support for a broad range of models hosted on the Hugging Face Hub, facilitating seamless integration and reproducibility within our setup. Models were served through the Chat Completions API and run with default decoding parameters, except for `temperature` = 0 to ensure deterministic generation. Each document image was submitted as a single request and evaluated under the two prompting conditions defined in Table 7 (simple instruction and Chain-of-Thought prompt), ensuring consistency with the commercial LLM protocol.

5.3.4. Prompting strategy

General-purpose multimodal LLMs (GPT-4o, Gemini 2.0 Flash, Claude Haiku 4.5, Gemma 3, and Qwen2.5-VL) were evaluated under two prompting conditions: (*i*) a simple instruction prompt, and (*ii*) a Chain-of-Thought (CoT) prompt that guides the model through an explicit multi-step procedure, analyzing layout and resolving ambiguous characters, before returning only the final transcription. Document-specialized models were prompted using task-specific instructions aligned with their dedicated inference APIs. The prompts adopted for each system category are summarized in Table 7.

5.3.5. Open-source OCR engines

Open-source OCR engines were accessed via their official Python libraries using default configurations and pre-trained weights. In our experiments, we employed version 5.0 of *Tesseract* (Smith, 2007) with default settings through the `pytesseract` interface. *DocTR* was executed using its official Python package with the PyTorch backend (Paszke et al., 2019) and GPU acceleration enabled. *PaddleOCR* was run through the `paddleocr` library, built on the PaddlePaddle deep-learning framework, and installed in both CPU and GPU configurations to assess performance differences across hardware setups. Transcriptions were generated independently for each input file. For all systems, the default configurations provided by the respective library versions listed in Table 1 were used.

5.3.6. Fine-tuning procedure

PaddleOCR offers a comprehensive toolkit for both training and evaluation, providing flexible configuration files, data preprocessing utilities, and pretrained models for a variety of OCR tasks. To assess the benefits of domain adaptation, we fine-tuned the SVTR-LCNet recognition module of PaddleOCR on each dataset. Ground-truth annotations were reformatted to match the requirements of PaddleOCR’s training pipeline, and three separate models were trained: one for SROIE, one for FUNSD, and one for IAM. Fine-tuning was initialized from the official PP-OCRv3 pretrained weights, using the default training configurations provided by the library.[11] Fine-tuning was performed for 50 epochs on the hardware setup previously described.

All experiments were orchestrated through reproducible Python scripts, available in the released repository.

[11] https://github.com/PaddlePaddle/PaddleOCR/blob/main/configs/rec/PP-OCRv3/PP-OCRv3_mobile_rec.yml

5.4. Evaluation metrics

The evaluation of OCR systems requires considering multiple aspects of performance to provide insights of practical relevance. In this study, we decided to focus on three key factors: transcription accuracy, processing time, and costs. Together, these metrics provide a balanced and extensive framework for the scientific validity of different OCR solutions, as well as their practical suitability and impact in industrial applications.

Transcription accuracy. Regardless of the underlying architecture, the fundamental indicator of OCR quality is the fidelity of the transcribed text. In our evaluation, transcription accuracy is assessed at both the character and word levels using *Character Error Rate* (CER) and *Word Error Rate* (WER) (Rice et al., 1993). Both metrics are derived from the edit distance (Levenshtein distance) (Kanai et al., 1993), which quantifies the minimum number of elementary operations (insertions, deletions, and substitutions) required to transform the output provided by the OCR system into the ground truth text:

$$CER = \frac{N_{ins} + N_{del} + N_{sub}}{N_{gt\_chars}} \qquad WER = \frac{N_{ins} + N_{del} + N_{sub}}{N_{gt\_words}} \tag{1}$$

Here, N_{gt_chars} and N_{gt_words} denote the total number of characters and words, respectively, in the ground truth text, while N_{ins}, N_{del}, and N_{sub} represent the counts of insertions, deletions, and substitutions at the respective level.

Transcription metrics were computed using the ISRI Analytic Tools for OCR Evaluation (Rice et al., 1993), a well-established framework for OCR benchmarking. More specifically, we employed `ocreval`,[12] a modern UTF-8–compatible reimplementation of the ISRI toolkit (Santos, 2019), which reproduces the original evaluation methodology for contemporary OCR systems. To directly capture both the accuracy of individual character transcription and the ability to produce correct word units, we report *Character Accuracy (CA)* and *Word Accuracy (WA)*, derived following the standard formulations of CER and WER.

$$CA = 1 - CER \qquad WA = 1 - WER \tag{2}$$

It is worth observing that CA and WA, being based on full-page linearized text, are sensitive to the serialization order in which text regions are concatenated into a single string. This is particularly relevant for structured documents such as FUNSD, where ground-truth annotations follow a specific field ordering that may differ from the output order produced by a given system, and for multi-column documents such as those in the FOX dataset, where reading order detection is an additional source of variability. A system that correctly recognizes all text but outputs it in a different order will be unfairly penalized by these metrics.

To address this limitation, we additionally report *Flexible Character Accuracy* (FCA) (Clausner et al., 2020), an edit-distance-based measure specifically designed to be invariant to reading order. FCA compares substrings of the recognized and ground-truth text in a flexible way, effectively decoupling character recognition accuracy from layout analysis and serialization errors (Clausner et al., 2020). This makes it particularly suitable for evaluating systems on complex layout documents, where reading order cannot be assumed to be consistent across systems. As FCA operates on line units, for OCR engines and OCR-oriented systems that natively return line-level transcriptions, we used the provided results. For multimodal LLMs that return plain text, we derived line units from newline characters (`\n`) in the generated outputs.

We further report *Normalized Edit Distance* (NED) (Marzal & Vidal, 2002) as an additional character-level measure that complements CA by relaxing the strictness of reference-length normalization. Let s_{pred} and s_{gt} denote the predicted and ground-truth character sequences, with lengths $|s_{pred}|$ and $|s_{gt}|$, respectively: NED is defined using the standard edit distance formulation, with unit costs for insertions, deletions, and substitutions.

$$NED(s_{pred}, s_{gt}) = \frac{N_{ins} + N_{del} + N_{sub}}{max(|s_{pred}|, |s_{gt}|)} \tag{3}$$

Unlike CER, which normalizes by the reference length alone, this normalization mitigates the tendency to over-penalize cases where the prediction is substantially longer (or shorter) than the ground truth (Marzal & Vidal, 2002). For consistency with our accuracy-oriented reporting, the NED is converted into a similarity score:

$$NED_{sim}(s_{pred}, s_{gt}) = 1 - NED(s_{pred}, s_{gt}) = 1 - \frac{N_{ins} + N_{del} + N_{sub}}{max(|s_{pred}|, |s_{gt}|)} \tag{4}$$

In this form, higher values indicate more similar strings, providing a length-robust view of transcription accuracy that is especially informative when systems differ in their propensity for omissions versus insertions. Together, CA, WA, FCA, and NED_{sim} provide a comprehensive and complementary view of transcription accuracy, accounting for both character-level fidelity and robustness to layout-induced serialization variability.

To ensure consistency in the accuracy evaluation, all model outputs were normalized before metric computation. For the English datasets (SROIE, FUNSD, IAM and the English subset of Fox), we applied a conservative character whitelist to reduce the impact of minor formatting inconsistencies across systems (e.g., rare symbols, or non-standard punctuation) that are marginal to OCR quality. Concretely, we retained uppercase and lowercase Latin letters (A–Z, a–z), digits (0–9), whitespace, and standard punctuation marks (`. , ; : ! ? ( ) [ ] { } / \ @ # – + _ ’ "`).

For the Chinese subset of FOX, we evaluate recognition directly at the character level on the original Chinese Unicode text. We report character-level metrics (CA, FCA, and NED) and omit word-level metrics because Chinese does not provide explicit whitespace-delimited word boundaries.

[12] https://github.com/eddieantonio/ocreval

Processing time. Another key operational factor in the assessment of OCR pipelines is their time efficiency, that directly affects the feasibility of deploying OCR systems in high-throughput and real-time scenarios.

Our evaluation accounts for the different nature of processing time across system types: for commercial OCR engines and multimodal LLMs, which are accessed through cloud APIs, processing time includes both inference time and network latency. In contrast, for open-source systems, executed locally, processing time corresponds solely to the model's inference time. For each dataset and system, we measured the time (in seconds) required to process a single page and report summary statistics (mean, median, minimum, and maximum), complemented with confidence intervals to capture variability.

Monetary costs. In addition to accuracy and processing time, we also considered the cost of running OCR systems as a third critical factor. Costs typically encompass both computational resources and monetary expenses, and become particularly relevant when deploying document processing pipelines at scale. In this study, we focused specifically on monetary costs, which vary depending on the type of technology employed:

- Commercial OCR services generally follow a *page-based* pricing model.
- Multimodal LLMs adopt a *token-based* pricing model, where both input and output tokens contribute to the total cost.
- Open-source solutions do not incur direct monetary charges; their cost is estimated based on the infrastructural resources required to run the experiments.

It is important to note that this analysis does not aim to suggest cost-optimization strategies but rather to report the expenses effectively incurred during experimentation, which differ across system categories.

6. Results

This section presents the empirical results of all evaluated systems, structured by scenario: printed receipts (SROIE), scanned forms (FUNSD), free-text handwritten (IAM) and dense-text documents in English and Chinese (FOX). Each scenario is analyzed in terms of transcription accuracy, followed by separate discussions on speed and cost.

6.1. Transcription accuracy

In the following, we present the transcription accuracy results for all evaluated systems across the four datasets. For each system and dataset, results are reported as 95% confidence intervals for the mean of each metric, computed from the sample mean and standard deviation across all evaluated pages using Student's t distribution ($t_{0.975;n-1}$). This choice supports statistically grounded comparisons between systems, capturing both average performance and variability across documents, and helping to discriminate consistent behavioral differences from mere noisy fluctuations.

Table 8 summarizes results on the SROIE dataset, which consists of printed receipts with structured and linear layouts. Overall, performance is high across most solutions, though with notable differences within categories. Commercial OCR services dominate the ranking, with *Amazon Textract* and *Azure Vision* achieving peak mean accuracies with narrow confidence intervals on all metrics, exhibiting highly reliable behavior across the dataset. Within this category, a notable pattern emerges for *Google Vision API* and *Google Document AI*, which show lower CA ($\approx$ 90.00) compared to FCA ($\approx$ 97.50): this large gap indicates that a fraction of their errors stems from reading order inconsistencies rather than character misrecognition, even on a dataset with relatively simple layout structure.

Among open-source OCR systems, *DocTR* emerges as the strongest performer and remains competitive with the best commercial services, particularly at the character level. *PaddleOCR* benefits substantially from fine-tuning, with performance gains of 4 (character level) and 20 (word level) over the pre-trained version; conversely, *Tesseract* lags behind all other systems. From a qualitative error analysis driven by the marginal CA-FCA discrepancies, we observed that errors are mainly due to standard mistranscriptions and limited coverage issues (e.g., small omissions).

Commercial multimodal LLMs form a tight cluster in terms of mean performance, with *Mistral Document AI* as the strongest model closely followed by the others. A noteworthy observation concerns *Gemini 2.0 Flash*, which shows a relatively large FCA-CA gap (5), suggesting that even without CoT its character recognition is stronger than CA alone indicates, with errors partly attributable to reading order variability.*Claude Haiku 4.5 without CoT* presents a distinctive anomaly: a low CA (82.73) paired with a high WA (92.36), pointing to errors concentrated at the character level within otherwise correctly recognized words. The effect of Chain-of-Thought prompting is positive for *Claude Haiku 4.5* and *Gemini 2.0 Flash*, while *GPT-4o* shows a marginal and mixed effect.

Open-source LLMs display the largest performance spread. *Qwen2.5-VL-3B* achieves a surprisingly strong CA (94.43) and FCA (96.38), approaching commercial LLMs with a modest FCA-CA gap ($\approx$ 2.00), consistent with reliable reading order handling. At the opposite extreme, *Gemma 3-4B without CoT* records the lowest CA and WA overall, together with one of the largest FCA-CA gap in the table ($\approx$ 13.7). This pattern indicates that its failures are not dominated by reading-order inconsistencies alone. Qualitative error analysis revealed recurrent repetition loops and layout-driven transcription hallucinations, which artificially inflate edit operations and depress CA even when some local fragments are correctly recognized. CoT substantially recovers its performance (+19.62 CA, +11.03 FCA). *DeepSeek-OCR* shows an even larger FCA-CA gap ($\approx$ 16.2) despite a relatively high WA (91.17), suggesting that many word-level units are reproduced correctly but serialized in an order that diverges markedly from the ground truth; here as well, we observed occasional repetition loops and the tendency to verbalize non-textual layout cues.

Table 8
Accuracy metrics on the SROIE dataset (printed receipts). Bold highlights the best absolute score per metric; underlined highlights the best in its category.

Category	Solution	CA	WA	NED_{sim}	FCA
Open-source OCR	Tesseract	(79.12 ± 2.33)	(70.17 ± 2.33)	(79.96 ± 2.20)	(82.20 ± 2.33)
	DocTR	(95.74 ± 0.37)	(91.45 ± 0.76)	(95.83 ± 0.36)	(97.08 ± 0.28)
	PaddleOCR	(89.56 ± 0.55)	(67.68 ± 1.45)	(89.57 ± 0.55)	(91.55 ± 0.49)
	PaddleOCR-FT	(93.86 ± 0.47)	(87.81 ± 0.88)	(93.91 ± 0.47)	(95.75 ± 0.36)
Commercial OCR	Azure Vision	(96.17 ± 0.39)	(94.03 ± 0.50)	(96.28 ± 0.35)	(97.80 ± 0.30)
	Azure Document Intelligence	(94.99 ± 0.45)	(93.48 ± 0.49)	(95.12 ± 0.42)	(97.64 ± 0.31)
	Google Vision API	(89.97 ± 0.74)	(90.90 ± 0.60)	(90.14 ± 0.72)	(97.49 ± 0.31)
	Google Document AI	(90.03 ± 0.75)	(90.91 ± 0.60)	(90.20 ± 0.72)	(97.52 ± 0.30)
	Amazon Textract	**(96.89 ± 0.36)**	**(95.00 ± 0.41)**	**(96.97 ± 0.33)**	**(98.11 ± 0.25)**
Commercial LLMs	Claude Haiku 4.5	(82.73 ± 1.66)	(92.36 ± 0.79)	(85.80 ± 1.27)	(90.77 ± 1.34)
	Claude Haiku 4.5-CoT	(92.40 ± 1.22)	(92.79 ± 0.64)	(93.22 ± 0.98)	(95.81 ± 1.05)
	GPT-4o	(92.13 ± 1.11)	(94.21 ± 0.47)	(93.13 ± 0.83)	(94.06 ± 1.09)
	GPT-4o-CoT	(91.24 ± 1.04)	(94.16 ± 0.47)	(92.25 ± 0.78)	(94.49 ± 1.00)
	Gemini 2.0 Flash	(92.52 ± 0.71)	(91.84 ± 0.63)	(92.66 ± 0.68)	(97.49 ± 0.28)
	Gemini 2.0 Flash-CoT	(93.98 ± 0.55)	(94.12 ± 0.81)	(94.16 ± 0.52)	(97.69 ± 0.27)
	Mistral Document AI	(94.18 ± 0.50)	(94.48 ± 0.42)	(94.45 ± 0.45)	(95.84 ± 0.42)
Open-source LLMs	Qwen2.5-VL-3B	(94.43 ± 0.71)	(92.99 ± 0.65)	(94.59 ± 0.67)	(96.38 ± 0.63)
	Qwen2.5-VL-3B-CoT	(92.54 ± 1.18)	(92.27 ± 1.01)	(92.93 ± 1.11)	(95.90 ± 1.13)
	Gemma 3-4B	(69.94 ± 1.43)	(73.37 ± 1.41)	(70.52 ± 1.38)	(83.67 ± 1.14)
	Gemma 3-4B-CoT	(89.56 ± 0.93)	(88.45 ± 0.88)	(89.91 ± 0.85)	(93.52 ± 0.78)
	PaddleOCR-VL	(89.30 ± 1.67)	(88.57 ± 1.03)	(89.62 ± 1.50)	(92.14 ± 1.64)
	DeepSeek-OCR	(75.79 ± 1.93)	(91.17 ± 0.99)	(80.07 ± 1.62)	(92.01 ± 1.41)

Table 9
Accuracy metrics on the IAM dataset (handwritten text). Bold highlights the best absolute score per metric; underlined highlights the best in its category.

Category	Solution	CA	WA	NED_{sim}	FCA
Open-source OCR	Tesseract	(52.19 ± 2.14)	(19.78 ± 2.00)	(52.22 ± 2.14)	(52.36 ± 2.12)
	DocTR	(70.03 ± 1.57)	(37.34 ± 2.51)	(70.31 ± 1.57)	(70.60 ± 1.53)
	PaddleOCR	(45.58 ± 2.54)	(20.37 ± 2.21)	(45.58 ± 2.54)	(46.29 ± 2.58)
	PaddleOCR-FT	(83.99 ± 1.24)	(72.16 ± 1.92)	(84.39 ± 1.17)	(85.64 ± 1.05)
Commercial OCR	Azure Vision	(93.31 ± 0.75)	(94.46 ± 0.66)	(93.55 ± 0.71)	(93.45 ± 0.74)
	Azure Document Intelligence	(92.88 ± 0.81)	(94.19 ± 0.75)	(93.12 ± 0.77)	(93.11 ± 0.77)
	Google Vision API	(88.58 ± 1.42)	(88.95 ± 1.13)	(88.78 ± 1.39)	(93.32 ± 0.92)
	Google Document AI	(88.74 ± 1.31)	(89.14 ± 1.07)	(88.91 ± 1.28)	(93.34 ± 0.90)
	Amazon Textract	(93.68 ± 0.77)	(87.07 ± 1.25)	(93.70 ± 0.77)	(93.85 ± 0.77)
Commercial LLMs	Claude Haiku 4.5	(91.19 ± 0.92)	(93.41 ± 0.84)	(91.60 ± 0.88)	(91.49 ± 0.91)
	Claude Haiku 4.5-CoT	(94.72 ± 0.84)	(93.40 ± 0.82)	(94.78 ± 0.83)	(94.87 ± 0.83)
	GPT-4o	(94.00 ± 2.37)	(93.95 ± 2.56)	(94.02 ± 2.37)	(94.16 ± 2.37)
	GPT-4o-CoT	(95.48 ± 1.85)	(95.68 ± 2.00)	(95.52 ± 1.85)	(95.66 ± 1.85)
	Gemini 2.0 Flash	(94.50 ± 0.91)	(90.01 ± 1.26)	(94.53 ± 0.90)	(95.01 ± 0.81)
	Gemini 2.0 Flash-CoT	(96.57 ± 0.70)	(95.13 ± 0.81)	(96.59 ± 0.70)	(96.76 ± 0.70)
	Mistral Document AI	**(97.29 ± 0.69)**	**(96.97 ± 0.55)**	**(97.31 ± 0.69)**	**(97.46 ± 0.69)**
Open-source LLMs	Qwen2.5-VL-3B	(96.94 ± 0.35)	(96.15 ± 0.52)	(96.96 ± 0.35)	(97.07 ± 0.34)
	Qwen2.5-VL-3B-CoT	(96.79 ± 0.36)	(96.02 ± 0.53)	(96.81 ± 0.36)	(96.95 ± 0.33)
	Gemma 3-4B	(87.64 ± 1.09)	(92.52 ± 0.92)	(88.79 ± 0.97)	(87.99 ± 1.04)
	Gemma 3-4B-CoT	(95.09 ± 0.57)	(93.10 ± 0.70)	(95.19 ± 0.53)	(95.20 ± 0.56)
	PaddleOCR-VL	(95.99 ± 0.41)	(93.56 ± 0.72)	(96.01 ± 0.41)	(96.12 ± 0.40)
	DeepSeek-OCR	(93.50 ± 1.38)	(91.94 ± 1.00)	(93.56 ± 1.35)	(93.64 ± 1.38)

Takeaway (SROIE). On printed receipt-style documents, dedicated OCR engines are the most accurate and stable solutions, with commercial services dominating the ranking. Among multimodal LLMs, CoT prompting yields consistent accuracy gains, suggesting suggesting that the more structured prompt helps mitigate qualitative error patterns such as repetition loops and layout-driven hallucinations that particularly affect open-source models.

The FCA–CA divergence observed for several systems, most markedly for *DeepSeek-OCR* and *Gemma 3*, reveals that a non-trivial fraction of errors on this dataset stems from reading order inconsistencies and generative artefacts rather than character misrecognition.

Table 9 reports results on the IAM dataset, which consists of free-text handwritten documents, a setting fundamentally different from printed documents and where generalization capabilities matter more than layout-specific optimization. On this dataset, multimodal LLMs reverse the trend observed on SROIE, clearly outperforming dedicated OCR engines.

Among commercial LLMs, *Mistral Document AI* achieves the best overall results (CA: 97.29, WA: 96.97, FCA: 97.46) with tight confidence intervals, indicating both high accuracy and stable behavior. *Gemini 2.0 Flash-CoT* follows closely (CA: 96.57), while *GPT-4o-CoT* (CA: 95.48) remains competitive despite wider confidence intervals. The FCA-CA gap is minimal across this entire category (below 0.5), confirming that on IAM's linear layout reading order is not a source of error for any of these models. CoT prompting is consistently beneficial.

Also open-source LLMs perform remarkably well on this dataset. *Qwen2.5-VL-3B* achieves CA 96.94 and FCA 97.07, the best open-source result and competitive with Mistral, with the narrowest confidence intervals in its category, indicating highly stable behavior. Notably, CoT provides no benefit for *Qwen2.5-VL-3B*, suggesting the model already handles handwritten text effectively without the additional structured prompt. Conversely, *Gemma 3-4B* benefits enormously from CoT (+7.45 CA), the largest CoT gain in this category. *PaddleOCR-VL* (CA: 95.99) and *DeepSeek-OCR* (CA: 93.50) also perform well, with minimal FCA-CA gaps across the board.

Open-source OCR engines perform poorly without domain adaptation, with CA ranging from 45.58 (*PaddleOCR*) to 70.03 (*DocTR*). Critically, the FCA-CA gap is negligible for all systems in this category, confirming that their errors are genuine recognition failures rather than serialization artefacts, consistent with the linear, single-column layout of IAM pages. *PaddleOCR-FT* demonstrates the high impact of fine-tuning, recovering from CA 45.58 to 83.99 (+38.40), though it still falls well short of commercial OCR and LLM-based solutions, highlighting the limits of open-source OCR solutions in this scenario. In contrast, commercial OCR services achieve strong results, with *Amazon Textract* (CA: 93.68) and *Azure Vision* (CA: 93.31) leading the category.

Takeaway (IAM). On handwritten documents, multimodal LLMs reverse the trend observed on SROIE, clearly outperforming dedicated OCR engines in both accuracy and stability. CoT prompting is consistently beneficial across commercial and open-source models. The negligible FCA–CA gaps across all categories confirm that on IAM's linear layout reading order is not a source of error, and that performance differences reflect genuine recognition capabilities rather than serialization artefacts. Fine-tuning substantially recovers open-source OCR performance, yet even adapted models fall short of the top multimodal LLMs, highlighting the structural advantage of vision–language representations for unstructured, high-variability inputs.

Table 10 reports accuracy results on FUNSD, a dataset of low-resolution scanned forms with noisy acquisition artefacts and highly irregular, field-based layouts. It is the most challenging benchmark in our study: confidence intervals are markedly wider than elsewhere, indicating strong instance-to-instance variability and limited generalization across heterogeneous form templates.

Commercial OCR services provide the most reliable operating point. *Amazon Textract* attains the best character-centric scores (CA: 86.01 ± 2.55, NED_{sim}: 86.36 ± 2.43) and the highest FCA (94.10 ± 1.57), with comparatively tight intervals given the dataset difficulty; *Azure* services follow closely. Among non-OCR approaches, *GPT-4o-CoT* is the only system that surpasses *Amazon Textract* on a single metric, achieving the best WA overall (WA: 87.22 ± 2.06), while remaining below the top commercial OCR engines on CA/NED_{sim}.

A salient cross-family pattern is the systematic enlargement of the FCA–CA gap, confirming that FUNSD penalizes serialization on form-like layouts. This effect is already visible for open-source OCR (e.g., *DocTR*, CA: 83.71 ± 2.63 vs. FCA: 91.11 ± 1.79) and becomes extreme for *Google* services, where CA stays around 72 while FCA exceeds 93 (gap $\approx 21.00\%$), strongly suggesting that many fragments are recognized but concatenated in an order that diverges from the ground-truth convention for fields.

Comparing simple and CoT prompt strategies clarifies the role of the structured prompt under noisy layouts. Chain-of-Thought is particularly beneficial for models prone to severe failures: *Gemma 3-4B* improves sharply in CA (48.39→73.35), consistent with a reduction of omissions and structural breakdowns. *Gemini 2.0 Flash* also benefits substantially in CA/NED_{sim} (71.32→80.81), with a notable reduction in variability, while FCA remains very high (≈ 93.00), suggesting improved local robustness without fully resolving reading-order ambiguity. *GPT-4o* shows only a modest CA gain but a sizeable WA increase (83.58→87.22), indicating improved word-level exactness. In contrast, *Qwen2.5-VL-3B* changes little across variants.

The *PaddleOCR* vs. *PaddleOCR-FT* comparison shows that fine-tuning does not uniformly transfer to FUNSD: while WA improves markedly (65.52→76.98), CA and FCA slightly decrease within overlapping intervals.

It is worth noting that FUNSD is the dataset where we observe the highest incidence of non-standard failure modes, which plausibly contributes to the lower character-level scores and broader intervals. These include graphic-element hallucinations (e.g., delimiter-like sequences), repetition loops (notably for *PaddleOCR-VL*), severe and partial omissions (e.g., *Gemma 3 without CoT*; *Tesseract* and *DeepSeek-OCR*), boilerplate preambles in several LLM outputs, and occasional schema-mismatched responses (e.g., *Gemini 2.0 Flash* returning bounding-box lists with coordinates). Collectively, these behaviors emphasize that, on scanned forms, performance depends not only on recognition quality but also on controlling serialization, format adherence, and layout-induced degeneration modes.

Takeaway (FUNSD). FUNSD is the most challenging dataset in our study, with wide confidence intervals across all systems reflecting strong instance-to-instance variability on noisy scanned forms. Commercial OCR services provide the most reliable operating point. The systematic enlargement of the FCA–CA gap across all system categories confirms that FUNSD penalizes serialization order more than any other dataset in our benchmark, with the most extreme cases observed for *Google* services (gap ≈21). CoT prompting is particularly effective for models prone to structural failures, with the largest gains observed for *Gemma 3* (+24.96 CA) and *Gemini 2.0 Flash* (+9.49 CA). Fine-tuning improves word-level accuracy for *PaddleOCR* but has limited impact, highlighting that domain adaptation alone is insufficient to address the structural challenges of form-based documents.

Table 11 reports accuracy metrics on the English subset of the FOX dataset, composed of text-dense documents with single-, double-, and mixed-column layouts. This setting introduces two distinct challenges: local recognition under high character density and, more importantly, global serialization under multi-column reading order ambiguity. As a consequence, several systems operate

Table 10
Accuracy metrics on the FUNSD dataset (scanned forms). Bold highlights the best absolute score per metric; underlined highlights the best in its category.

Category	Solution	CA	WA	NED_{sim}	FCA
Open-source OCR	Tesseract	(65.37 ± 4.75)	(54.22 ± 5.17)	(65.39 ± 4.76)	(71.86 ± 4.75)
	DocTR	(83.71 ± 2.63)	(78.83 ± 2.74)	(84.13 ± 2.48)	(91.11 ± 1.79)
	PaddleOCR	(81.48 ± 3.03)	(65.52 ± 4.39)	(81.49 ± 3.03)	(88.00 ± 2.65)
	PaddleOCR-FT	(79.43 ± 3.28)	(76.98 ± 3.44)	(80.25 ± 3.02)	(86.76 ± 2.15)
Commercial OCR	Azure Vision	(85.18 ± 2.47)	(84.86 ± 2.12)	(85.61 ± 2.32)	(93.46 ± 1.66)
	Azure Document Intelligence	(84.30 ± 3.05)	(84.14 ± 2.35)	(84.73 ± 2.95)	(93.35 ± 1.60)
	Google Vision API	(72.38 ± 3.82)	(76.95 ± 2.74)	(72.87 ± 3.75)	(93.44 ± 1.55)
	Google Document AI	(72.45 ± 3.74)	(77.20 ± 2.71)	(72.95 ± 3.66)	(93.41 ± 1.56)
	Amazon Textract	**(86.01 ± 2.55)**	(84.36 ± 2.33)	**(86.36 ± 2.43)**	**(94.10 ± 1.57)**
Commercial LLMs	Claude Haiku 4.5	(66.80 ± 6.45)	(86.03 ± 2.08)	(72.68 ± 4.44)	(78.63 ± 6.79)
	Claude Haiku 4.5-CoT	(69.73 ± 6.82)	(85.97 ± 1.95)	(75.22 ± 4.59)	(78.02 ± 7.10)
	GPT-4o	(77.55 ± 5.81)	(83.58 ± 2.26)	(79.73 ± 4.47)	(86.07 ± 6.04)
	GPT-4o-CoT	(78.54 ± 4.93)	**(87.22 ± 2.06)**	(80.67 ± 3.51)	(87.97 ± 4.92)
	Gemini 2.0 Flash	(71.32 ± 7.18)	(81.87 ± 2.76)	(74.83 ± 4.92)	(93.45 ± 1.55)
	Gemini 2.0 Flash-CoT	(80.81 ± 3.43)	(85.90 ± 2.72)	(81.25 ± 3.32)	(93.07 ± 2.50)
	Mistral Document AI	(75.83 ± 6.66)	(80.88 ± 4.49)	(78.49 ± 4.95)	(83.15 ± 7.02)
Open-source LLMs	Qwen2.5-VL-3B	(77.87 ± 4.34)	(81.68 ± 3.52)	(78.62 ± 4.06)	(85.12 ± 3.79)
	Qwen2.5-VL-3B-CoT	(77.50 ± 4.39)	(81.74 ± 3.39)	(78.40 ± 3.95)	(85.49 ± 3.81)
	Gemma 3-4B	(48.39 ± 7.04)	(55.86 ± 7.19)	(50.05 ± 6.79)	(71.69 ± 6.23)
	Gemma 3-4B-CoT	(73.35 ± 5.90)	(78.13 ± 3.88)	(74.97 ± 4.82)	(81.61 ± 5.61)
	PaddleOCR-VL	(69.47 ± 9.13)	(82.13 ± 4.59)	(71.41 ± 8.33)	(74.95 ± 9.58)
	DeepSeek-OCR	(66.18 ± 6.95)	(75.12 ± 5.75)	(67.83 ± 6.54)	(75.16 ± 7.40)

near ceiling on CA/WA/NED_{sim}, and the most informative discriminator becomes whether a model can preserve the ground-truth reading flow.

Open-source OCR engines reveal a sharp internal split on this dataset. *Tesseract* achieves a strong CA (96.34) with a modest FCA-CA gap (≈ 1.70), suggesting reliable reading order handling on dense-text pages. In contrast, *DocTR* and *PaddleOCR* show substantially lower CA (≈ 85.80) despite near-perfect FCA (≈ 99.0 with extremely narrow intervals), yielding higher FCA-CA gaps (≈ 13.00). This dissociation pairs with the evidence for both systems of recognizing characters with high fidelity but failing to serialize them in the correct reading order across columns.

A similar split is observable also in commercial OCR services. *Azure Document Intelligence* achieves the best CA in the category (97.97) with a minimal FCA-CA gap, closely followed by *Google Vision API* and *Google Document AI*. In contrast, *Azure Vision* and *Amazon Textract* show substantially lower CA (≈ 86.30) with a near-perfect FCA (≈ 99.5), similarly to *DocTR* and *PaddleOCR*. This confirms that layout analysis capability (and not recognition quality) is the primary differentiator among commercial OCR services on dense multi-column documents.

Commercial LLMs perform strongly on this dataset, with *Mistral Document AI* achieving the best values for CA and NED_{sim} and a near-best value for WA. Almost all the systems within this category exhibit tight FCA-CA gaps, reflecting the advantage of leveraging holistic layout understanding.

Among open-source LLMs, *PaddleOCR-VL* and *DeepSeek-OCR* demonstrate outstanding results, outperforming most of the others LLMs and matching the best commercial OCR services. *Gemma 3-4B* is the weakest system overall on this dataset. A qualitative inspection of its predictions highlighted the presence of errors related to omissions and boilerplate preambles presence, which reflect the sharp degrade in all the metrics. Unexpectedly, CoT further degrades *Gemma 3-4B* performance (≈ 5.40 CA), the largest negative CoT effect observed across the entire benchmark. We also observed sporadic omissions and the insertion of boilerplate preambles in other multimodal LLMs, most notably *GPT-4o* and *Qwen2.5-VL*. The difference is primarily on frequency and severity: for higher-performing models, these degeneration modes occur less often and typically affect shorter spans, which limits their impact on mean CA/WA and yields slightly tighter confidence intervals.

Takeaway (FOX-english). On dense-text English pages, the primary discriminator between systems is the ability to correctly serialize multi-column content in the ground-truth reading order. This is confirmed by the systematic FCA–CA dissociation observed for *DocTR*, *PaddleOCR*, *Azure Vision*, and *Amazon Textract*, which recognize characters with high fidelity but fail to reconstruct the correct column reading flow. Conversely, *Azure Document Intelligence* and LLM-based approaches show a clear advantage in this respect, with minimal FCA–CA gaps. Among open-source LLMs, *PaddleOCR-VL* and *DeepSeek-OCR* achieve outstanding results, matching the best commercial OCR services. CoT prompting shows mixed effects on this dataset: while beneficial for most models, it degrades *Gemma 3* performance, consistent with qualitative observations of increased omissions and boilerplate preamble insertions under structured CoT prompt on dense pages.

Table 12 reports results on the Chinese split of FOX.[13] Among commercial OCR services, *Azure Document Intelligence* and the two *Google* solutions define the top tier, with CA close to 98 and tight confidence intervals, indicating both high accuracy and stability.

[13] As discussed in Section 5.4, we omit word-level metrics because Chinese does not provide explicit whitespace-delimited word boundaries.

Table 11
Accuracy metrics on the FOX dataset (English subset; text-dense documents). Bold highlights the best absolute score per metric; underlined highlights the best in its category.

Category	Solution	CA	WA	NED_{sim}	FCA
Open-source OCR	Tesseract	(96.34 ± 1.17)	(92.87 ± 1.55)	(96.36 ± 1.16)	(98.01 ± 0.52)
	DocTR	(85.75 ± 5.11)	(87.38 ± 3.30)	(85.90 ± 5.06)	(98.80 ± 0.27)
	PaddleOCR	(85.81 ± 5.10)	(88.37 ± 3.32)	(85.92 ± 5.06)	(99.00 ± 0.22)
	PaddleOCR-FT	–	–	–	–
Commercial OCR	Azure Vision	(86.33 ± 5.08)	(89.56 ± 3.27)	(86.46 ± 5.04)	(99.47 ± 0.11)
	Azure Document Intelligence	(97.97 ± 0.99)	**(96.85 ± 0.80)**	(98.01 ± 0.97)	(99.49 ± 0.11)
	Google Vision API	(96.59 ± 1.77)	(96.07 ± 1.15)	(96.62 ± 1.75)	**(99.51 ± 0.10)**
	Google Document AI	(96.61 ± 1.77)	(96.06 ± 1.15)	(96.64 ± 1.75)	(99.50 ± 0.10)
	Amazon Textract	(86.27 ± 5.14)	(89.38 ± 3.31)	(86.39 ± 5.09)	(99.36 ± 0.15)
Commercial LLMs	Claude Haiku 4.5	(92.22 ± 3.65)	(91.63 ± 3.18)	(92.44 ± 3.58)	(95.34 ± 2.98)
	Claude Haiku 4.5-CoT	(93.71 ± 3.07)	(92.78 ± 2.48)	(93.96 ± 2.92)	(96.62 ± 2.29)
	GPT-4o	(97.33 ± 1.36)	(95.94 ± 0.99)	(97.49 ± 1.18)	(98.37 ± 0.99)
	GPT-4o-CoT	(95.32 ± 2.69)	(93.43 ± 2.62)	(95.35 ± 2.68)	(96.67 ± 2.53)
	Gemini 2.0 Flash	(97.42 ± 1.41)	(96.45 ± 1.12)	(97.44 ± 1.40)	(99.18 ± 0.67)
	Gemini 2.0 Flash-CoT	(97.29 ± 1.38)	(96.26 ± 1.10)	(97.32 ± 1.36)	(99.19 ± 0.66)
	Mistral Document AI	**(97.99 ± 0.97)**	(96.17 ± 0.89)	**(98.01 ± 0.95)**	(99.13 ± 0.20)
Open-source LLMs	Qwen2.5-VL-3B	(91.96 ± 3.36)	(90.25 ± 3.15)	(91.98 ± 3.35)	(93.47 ± 2.94)
	Qwen2.5-VL-3B-CoT	(94.73 ± 2.58)	(92.97 ± 2.36)	(94.79 ± 2.56)	(96.03 ± 2.11)
	Gemma 3-4B	(63.31 ± 4.00)	(64.12 ± 3.07)	(64.67 ± 3.41)	(65.07 ± 4.05)
	Gemma 3-4B-CoT	(57.96 ± 4.68)	(61.61 ± 3.21)	(60.99 ± 3.70)	(58.98 ± 9.18)
	PaddleOCR-VL	(97.30 ± 1.48)	(95.60 ± 1.36)	(97.35 ± 1.45)	(98.49 ± 1.06)
	DeepSeek-OCR	(97.92 ± 0.98)	(96.06 ± 0.89)	(97.93 ± 0.97)	(98.87 ± 0.22)

In contrast, Azure Vision drops to CA 89.23 while maintaining FCA 98.13, mirroring the reading order failures observed for this system on the English subset.

Within open-source OCR, only *PaddleOCR* is reported, and it exhibits a clear gap between CA (89.00 ± 5.19) and FCA (98.02 ± 0.37). As discussed before for *Azure Vision*, such a wide gap suggests that the model captures the correct character content, but failing to serialize it in the correct order on multi-column pages.

Commercial multimodal LLMs are markedly more heterogeneous and, for some models, substantially less stable. *Mistral Document AI* remains competitive (CA: 95.39 ± 0.64; FCA: 97.85 ± 0.64), approaching the strongest OCR pipelines. The most pronounced CoT effect is observed for *Gemini 2.0 Flash*, with improvements at the character-level accuracy and stability. Conversely, *GPT-4o* and *Claude Haiku 4.5* remain around CA ≈65–69 with wide intervals, indicating limited reliability on this dataset.

For open-source multimodal LLMs, *DeepSeek-OCR* achieves the strongest results (CA: 93.91; FCA: 96.15), followed by *Qwen2.5-VL-3B* and *PaddleOCR-VL* (CA ≈ 92 with FCA ≈ 94). CoT does not appear uniformly beneficial here: for instance, the CoT variant of *Qwen2.5-VL-3B* does not yield clear gains and even degrades NED_{sim}. *Gemma 3*'s poor performances (CA ≈12–14) are not meaningfully recovered by CoT, consistent with a fundamental difficulty for this model in handling dense Chinese document pages.

Qualitative error inspection enriches the quantitative analyses, revealing that the Chinese subset of FOX is one of the settings where non-standard transcription errors are frequent, especially among multimodal LLMs. In addition to the serialization issues already highlighted for OCR systems, we observed repetition loops in several models (*GPT-4o*, *Gemini 2.0 Flash*, *PaddleOCR-VL*), with *Gemma 3* exhibiting this behavior in more than half of the instances. Boilerplate preambles were also common, particularly for *Gemma 3* and *DeepSeek-OCR*. Omissions were observed for *PaddleOCR-VL* and *Qwen2.5-VL-3B*, typically affecting dense regions or column transitions and thereby degrading CA. Moreover, we recorded cases of schema drift, where *DeepSeek-OCR* occasionally produced outputs in a format different to a plain transcription, which is heavily penalized by alignment-based metrics despite containing text correctly transcripted. Finally, we recorded some cases in which *GPT-4o* refused to transcribe, returning generic messages (e.g., "Sorry, I can't assist you with that") which can be treated as major omissions, further reducing his reliability on this subset.

Takeaway (FOX-Chinese. This subset cleanly separates reliable systems from weak or unstable ones. The best results come from commercial OCR, led by *Azure Document Intelligence* and closely followed by the two *Google* solutions. Among multimodal LLMs, *Gemini 2.0 Flash-CoT* and *Mistral Document AI* are the strongest, while *DeepSeek-OCR* is the best open-source option. *Azure Vision* and *PaddleOCR* sit notably lower in CA despite near-ceiling FCA, indicating primarily serialization issues. The weakest models are *GPT-4o* and *Claude Haiku 4.5*, with *Gemma 3* collapsing to very low accuracy due to major generative errors.

6.2. Processing time

Processing time was analyzed separately for commercial and open-source solutions. For cloud-based systems, processing time was measured as the elapsed time between the submission of the API request and the receipt of the complete response, thus including both server-side inference time and network round-trip latency. To avoid rate limiting, consecutive requests were separated by

Table 12
Accuracy metrics on the FOX dataset (Chinese subset; dense-text documents). Bold highlights the best absolute score per metric; underlined highlights the best in its category.

Category	Solution	CA	WA	NED_{sim}	FCA
Open-source OCR	Tesseract	–	–	–	–
	DocTR	–	–	–	–
	PaddleOCR	(89.00 ± 5.19)	–	(94.83 ± 1.28)	(98.02 ± 0.37)
	PaddleOCR-FT	–	–	–	–
Commercial OCR	Azure Vision	(89.23 ± 5.24)	–	(94.88 ± 1.26)	(98.13 ± 0.38)
	Azure Document Intelligence	**(97.87 ± 0.40)**	–	(94.68 ± 1.29)	(98.08 ± 0.39)
	Google Vision API	(97.73 ± 0.49)	–	(94.69 ± 1.29)	**(98.21 ± 0.35)**
	Google Document AI	(97.72 ± 0.49)	–	(94.70 ± 1.29)	(98.20 ± 0.35)
	Amazon Textract	–	–	–	–
Commercial LLMs	Claude Haiku 4.5	(65.60 ± 4.78)	–	(94.76 ± 1.33)	(68.06 ± 4.6)
	Claude Haiku 4.5-CoT	(66.92 ± 4.80)	–	(94.87 ± 1.29)	(69.86 ± 4.56)
	GPT-4o	(66.15 ± 6.26)	–	**(95.50 ± 1.26)**	(67.22 ± 6.35)
	GPT-4o-CoT	(69.18 ± 6.14)	–	(95.45 ± 1.17)	(70.26 ± 6.23)
	Gemini 2.0 Flash	(84.19 ± 6.32)	–	(95.07 ± 1.21)	(85.33 ± 5.99)
	Gemini 2.0 Flash-CoT	(96.94 ± 1.85)	–	(94.66 ± 1.30)	(97.75 ± 1.63)
	Mistral Document AI	(95.39 ± 0.64)	–	(94.59 ± 1.31)	(97.85 ± 0.64)
Open-source LLMs	Qwen2.5-VL-3B	(92.10 ± 2.27)	–	(95.43 ± 1.13)	(94.50 ± 2.32)
	Qwen2.5-VL-3B-CoT	(91.86 ± 2.08)	–	(91.88 ± 2.08)	(93.87 ± 2.51)
	Gemma 3-4B	(11.72 ± 3.70)	–	(19.59 ± 3.21)	(12.93 ± 3.57)
	Gemma 3-4B-CoT	(13.73 ± 4.16)	–	(21.48 ± 3.61)	(17.80 ± 17.54)
	PaddleOCR-VL	(92.27 ± 3.37)	–	(94.97 ± 1.27)	(94.55 ± 3.46)
	DeepSeek-OCR	(93.91 ± 1.15)	–	(94.73 ± 1.27)	(96.15 ± 1.21)

fixed inter-request intervals, which were excluded from timing measurements. It should be noted that, because cloud-based services were queried remotely from our AWS instance, their reported processing times include network round-trip latency, transport-layer overhead, and service-side request handling in addition to model inference time. In contrast, open-source engines were timed during local execution on dedicated hardware. While this measurement protocol reflects realistic deployment conditions and end-to-end user experience, it may introduce a modest bias when comparing cloud-based services with locally executed systems. Accordingly, the reported latencies should be interpreted as end-to-end processing times rather than isolated model inference times.

Open-source systems were executed locally on the same hardware, ensuring controlled and reproducible conditions, though without production-level optimizations. Both CPU and GPU configurations were tested to quantify performance gaps and evaluate the feasibility of deployment without hardware acceleration.

In the following, for each system category, we report 95% confidence intervals for the mean processing time per page,[14] computed as described in 6.1.

6.2.1. Commercial solutions

Fig. 3 reports the collected processing time metrics on the SROIE dataset. Colored bars indicate mean values and vertical lines denote the confidence intervals. Plots are displayed on a logarithmic scale to improve readability, given the large latency differences between OCR services and multimodal LLMs.

Commercial OCR services are the fastest, with *Google Vision* averaging 0.29 s/page and *Azure Vision* 0.36 s/page. In contrast, multimodal LLMs are significantly slower overall: *Gemini 2.0 Flash* requires about 3.4 s/page, *Claude 4.5 Haiku* around 10 s/page, and *GPT-4o* approximately 12 s/page. Among multimodal models, *Mistral Document AI* achieves latency comparable to traditional OCR systems.

Consistent with the previous results, commercial OCR services are the fastest on the IAM dataset, as shown in Fig. 4. *Azure Vision* (0.37 s/page) and *Google Vision* (0.57 s/page) lead the group, while *Amazon Textract* and other document-oriented cloud solutions exhibit median latencies of approximately one second per page.

Among multimodal models, *Gemini 2.0 Flash* shows moderate latency (2.81 s median), whereas *Claude 4.5 Haiku* and *GPT-4o* are slower (4.14 s and 5.90 s median, respectively), yet still notably faster than in the SROIE scenario. This reduction reflects the lower token generation demand associated with the shorter text sequences in IAM compared to the longer and denser printed receipts in SROIE.

Results on the FUNSD dataset, reported in Fig. 5, are consistent with those obtained on the other datasets. Commercial OCR systems remain the fastest, with *Google Vision* (0.32 s/page) and *Azure Vision* (0.37 s/page) showing the lowest latencies. *Mistral Document AI* is notably efficient among next-generation multimodal solutions (1.06 s/page), achieving latency comparable to that of

[14] We report cost and processing-time for the simple-prompt condition only. The CoT prompt is operationally equivalent: in our setup it does not produce reasoning tokens and yields output of comparable length. Since cost and latency are driven by output tokens, the two conditions differ only in the slightly longer input prompt.

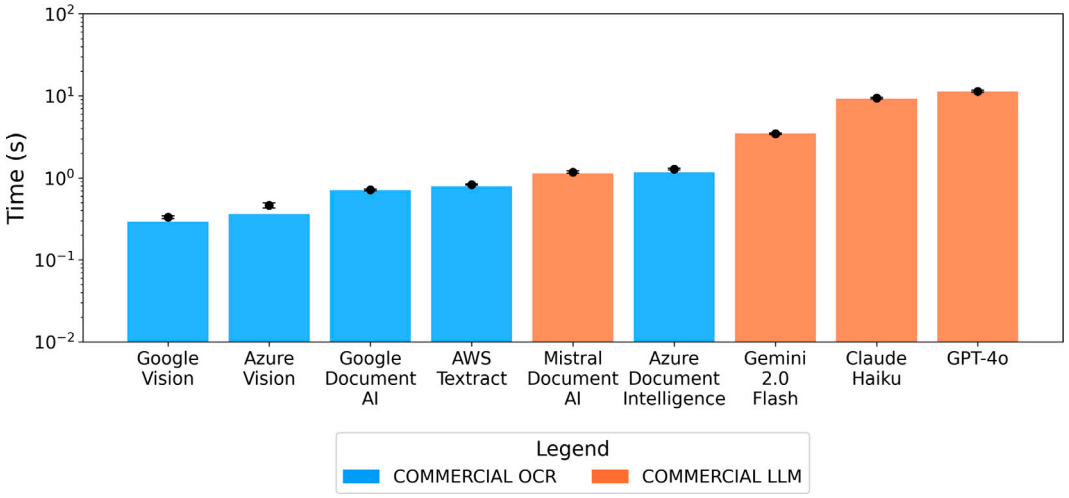

Fig. 3. Time processing metrics of Commercial OCR and LLM on SROIE.

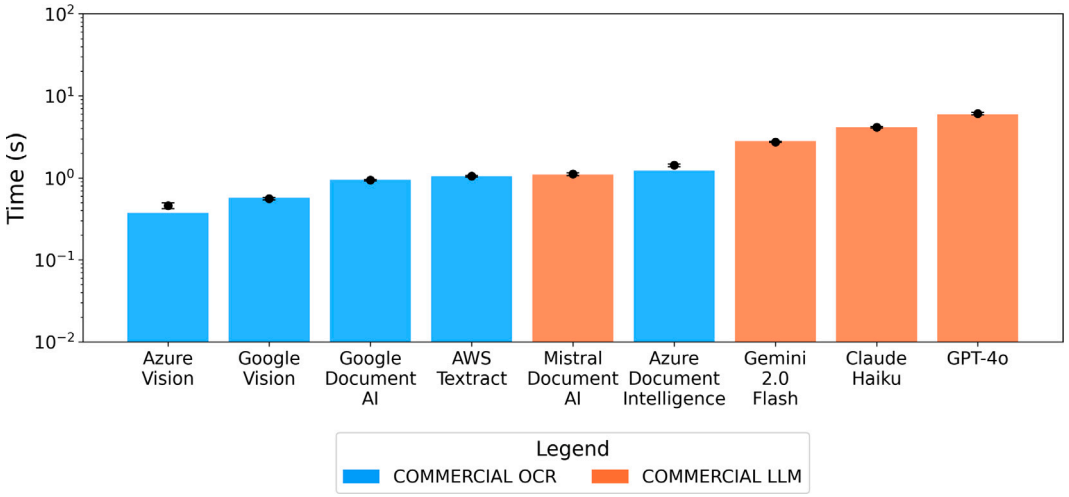

Fig. 4. Time processing metrics of Commercial OCR and LLM solutions on IAM.

commercial OCR engines such as *Amazon Textract* and *Google Document AI*. *Claude 4.5 Haiku* (7.83 s/page) and *GPT-4o* (10.56 s/page) are slower, reflecting the higher token generation cost associated with form-based layouts.

Results on the FOX dataset show even more pronounced latency differences across system categories, driven by the higher text density of its pages. As shown in Figs. 6 and 7, commercial OCR services remain the fastest tier: *Google Vision* leads with 0.48 s/page (EN) and 0.43 s/page (CN), while *AWS Textract* and *Azure Document Intelligence* are slower than in previous datasets, reflecting the greater complexity of dense multi-column layouts. *Mistral Document AI* again stands out among multimodal solutions, achieving 2.25 s/page (EN) and 2.54 s/page (CN), remaining competitive with document-oriented OCR services. General-purpose LLMs exhibit substantially higher latencies than observed on the previous datasets: *Gemini 2.0 Flash* requires approximately 5.82 s/page (EN) and 6.48 s/page (CN), while *Claude 4.5 Haiku* and *GPT-4o* reach 22.09 s/page and 24.15 s/page on the English subset, and 30.00 s/page and 26.53 s/page on the Chinese subset. These figures confirm that latency in generative models scales with output length, and that dense text pages impose a substantial generation overhead that makes such systems particularly unsuitable for high-throughput scenarios involving text-heavy documents.

6.2.2. *Open-source systems*

Figs. 8–12 summarize inference times across datasets for open-source OCR engines and multimodal LLMs.

On the SROIE dataset, *DocTR* and the GPU version of *PaddleOCR* achieve around 0.17 s per page, demonstrating real-time inference capability. *Tesseract*, operating solely on CPU, is considerably slower with a median latency of 1.88 s per page. Among open-source LLMs, *PaddleOCR-VL* and *DeepSeek-OCR* stand out with median latencies of 1.19 s and 1.49 s respectively, remaining

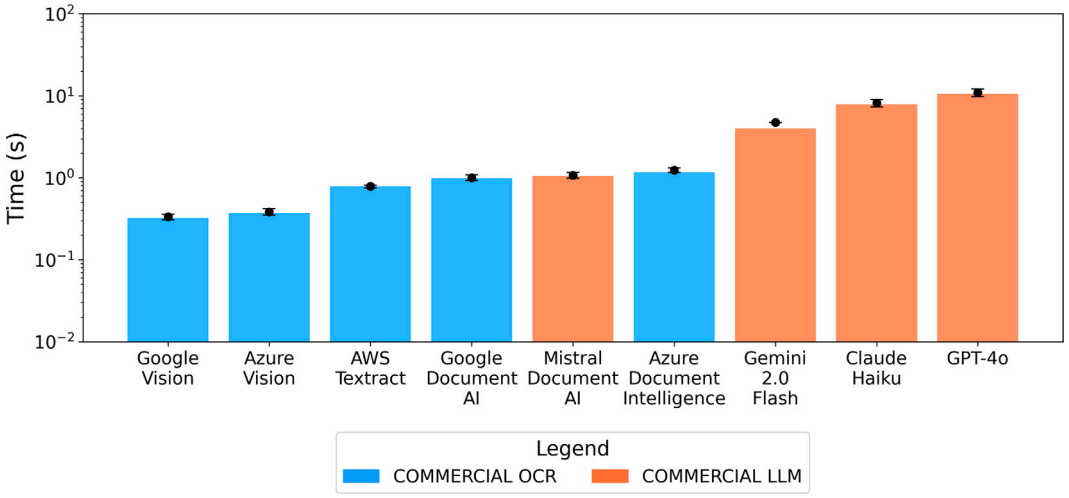

Fig. 5. Time processing metrics of Commercial OCR and LLM solutions on FUNSD.

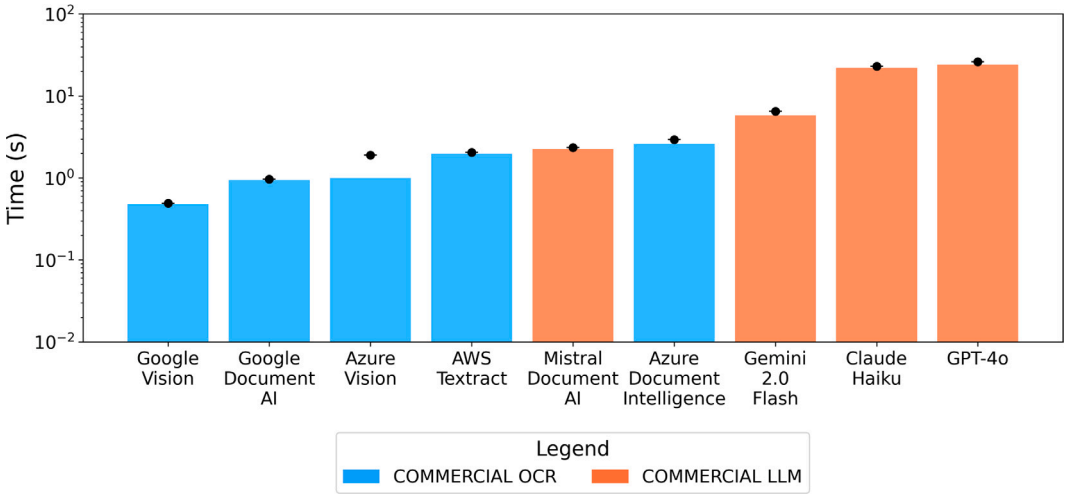

Fig. 6. Time processing metrics of Commercial OCR and LLM solutions on FOX (English subset).

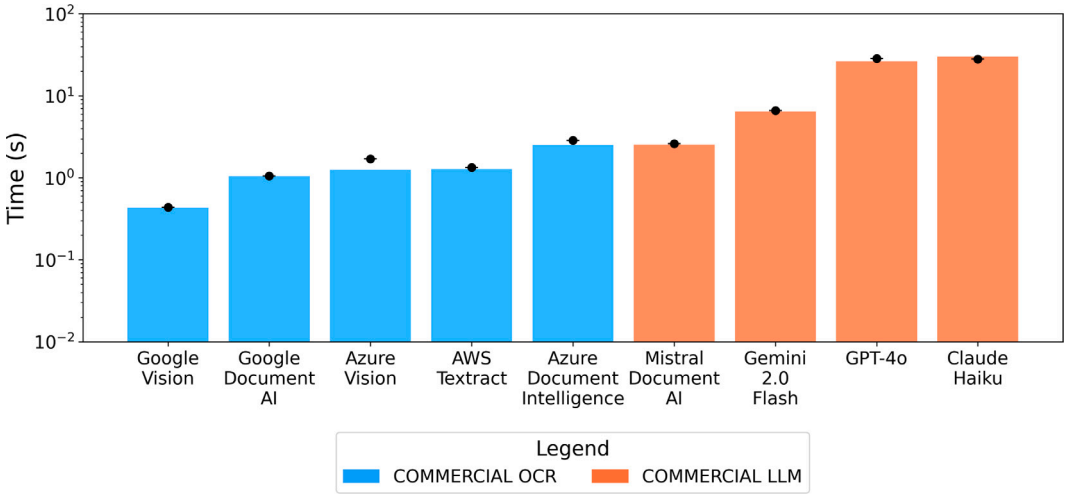

Fig. 7. Time processing metrics of Commercial OCR and LLM solutions on FOX (Chinese subset).

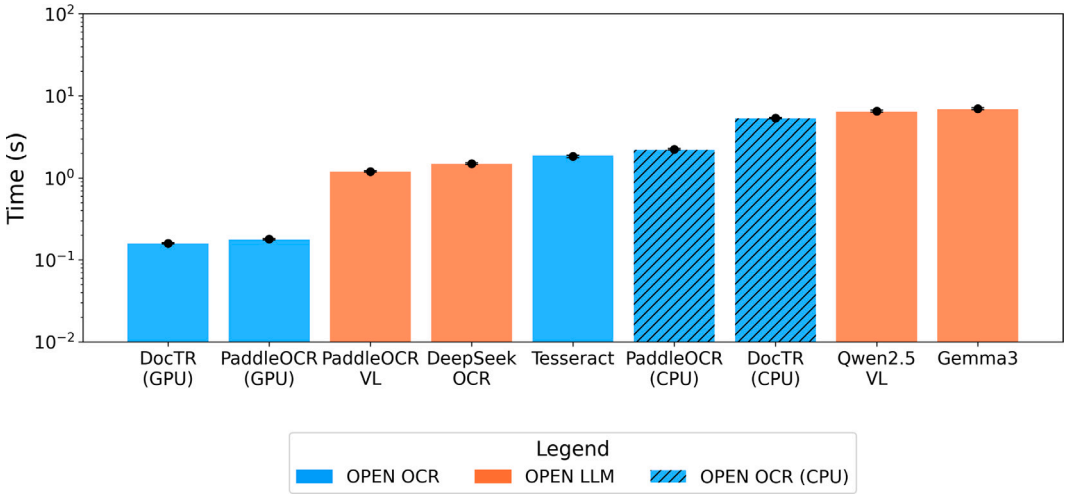

Fig. 8. Time processing metrics of open-source OCR and LLM solutions on SROIE.

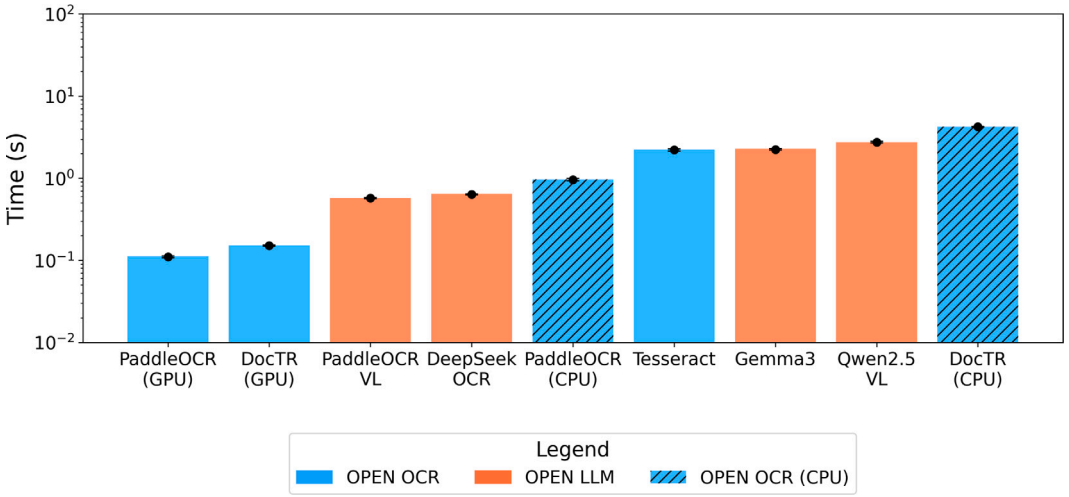

Fig. 9. Time processing metrics of open-source OCR and LLM solutions on IAM.

competitive with CPU-based OCR engines and substantially faster than general-purpose multimodal models. *Qwen2.5-VL (3B)* and *Gemma 3 (4B)*, running on GPU via the vLLM engine, require around six seconds per page (6.41 s and 6.83 s, respectively), confirming that general-purpose multimodal LLMs remain unsuitable for real-time scenarios without further optimization or more efficient serving infrastructures.

Inference times on the IAM dataset are generally lower and more stable across systems, reflecting the less text-dense nature of handwritten samples. *PaddleOCR* remains the most efficient engine, reaching 0.11 s per page on GPU and 0.97 s on CPU, while *DocTR* is slightly slower (0.15 s on GPU, 4.25 s on CPU). *PaddleOCR-VL* and *DeepSeek-OCR* achieve median inference times of 0.58 s and 0.65 s respectively, placing them among the fastest systems overall on this dataset, on par with a lightweight OCR engine as Tesseract. *Gemma 3 (4B)* and *Qwen2.5-VL (3B)* achieve median inference times of 2.28 s and 2.76 s, respectively. These results confirm that latency in multimodal architectures scales primarily with output length, whereas traditional OCR engines maintain stable inference times regardless of document type.

On the FUNSD dataset, latency distributions reflect the higher visual and structural complexity of form-like documents, which particularly affects generative models. *PaddleOCR* and *DocTR* running on GPU achieve nearly identical performance, with median latencies between 0.16 s and 0.19 s per page, confirming their scalability under GPU acceleration. *Tesseract* maintains a median of 1.47 s, offering stable but lower throughput compared to modern neural engines. *PaddleOCR-VL* and *DeepSeek-OCR* show higher variability on this dataset, with median latencies of 1.18 s and 1.32 s but wide confidence intervals, suggesting sensitivity to the structural complexity of form layouts. *Qwen2.5-VL (3B)* and *Gemma 3 (4B)* are considerably slower at 5.18 s and 6.61 s median, consistent with the pattern observed on SROIE. Across all experiments, commercial OCR engines consistently deliver

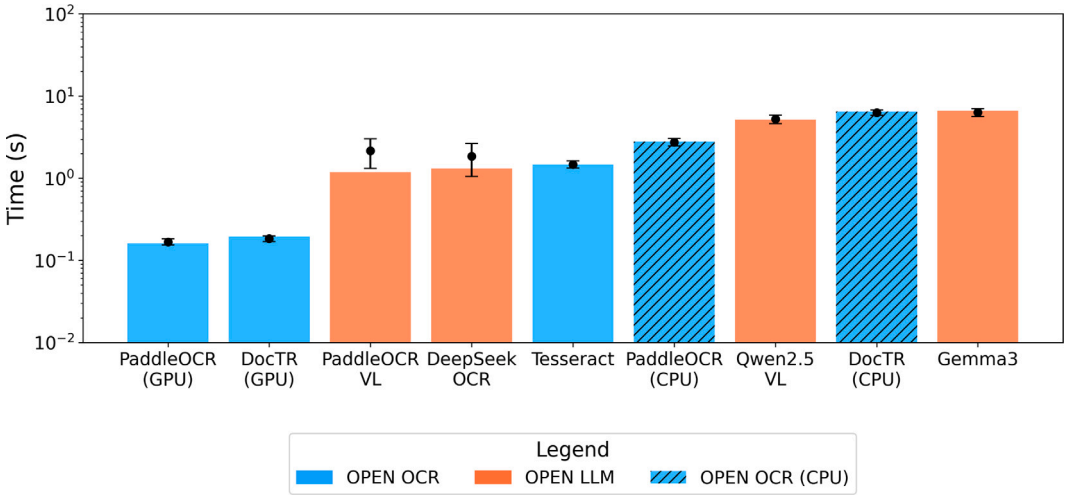

Fig. 10. Time processing metrics of open-source OCR and LLM solutions on FUNSD.

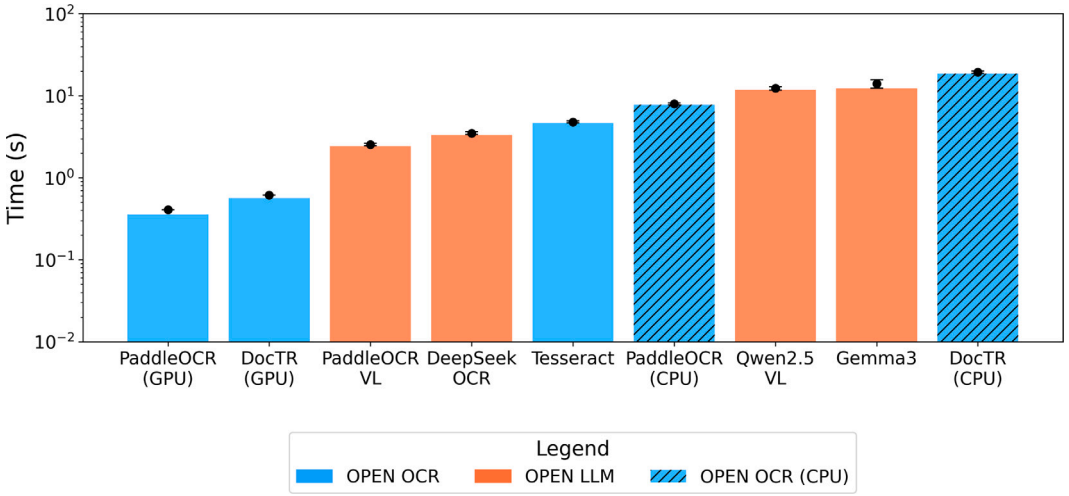

Fig. 11. Time processing metrics of open-source OCR and LLM solutions on FOX (English subset).

near-real-time performance, setting the benchmark for latency efficiency. Open-source systems, particularly *PaddleOCR* and *DocTR*, achieve comparable throughput under GPU acceleration, confirming their suitability for large-scale, production-grade deployments. The lightweight architecture of *PaddleOCR* further ensures competitive performance even in CPU-only environments, making it well-suited for resource-constrained document processing pipelines. Document-specialized open-source LLMs such as *PaddleOCR-VL* and *DeepSeek-OCR* occupy an interesting middle ground, achieving latencies close to CPU-based OCR engines while retaining the generalization capabilities of vision–language architectures. General-purpose multimodal LLMs remain one to two orders of magnitude slower, with latency influenced by sequence length and generation overhead.

On the English subset of FOX, GPU-accelerated OCR engines remain the fastest systems: *PaddleOCR* (GPU) achieves a median of 0.36 s/page and *DocTR* (GPU) 0.57 s/page. *Tesseract* and *PaddleOCR* on CPU rise substantially to 4.66 s and 7.79 s, reflecting the increased processing demand of dense multi-column layouts. *DocTR* on CPU reaches a median of 18.68 s, with a reduced sample size of 76 out of 112 pages, suggesting that a portion of jobs exceeded the allocated processing time.

Among open-source LLMs, *PaddleOCR-VL* and *DeepSeek-OCR* stand out with median latencies of 2.43 s and 3.32 s respectively, remaining well below general-purpose models. *Qwen2.5-VL (3B)* requires 11.89 s median, while *Gemma 3 (4B)* reaches 12.39 s with a substantially wider confidence interval (std 16.76 s), indicating high page-to-page variability likely driven by the density and layout complexity of FOX pages.

On the Chinese subset, PaddleOCR on GPU achieves a median of 0.75 s/page, while the CPU variant rises to 5.65 s/page. PaddleOCR-VL and DeepSeek-OCR maintain stable inference at 2.45 s and 3.46 s median, respectively. Qwen2.5-VL (3B) requires 13.33 s median, consistent with the English subset. Gemma 3 (4B), however, shows a severe latency degradation; this pattern suggests that the model struggles to terminate generation on Chinese dense-text, likely hitting output length limits.

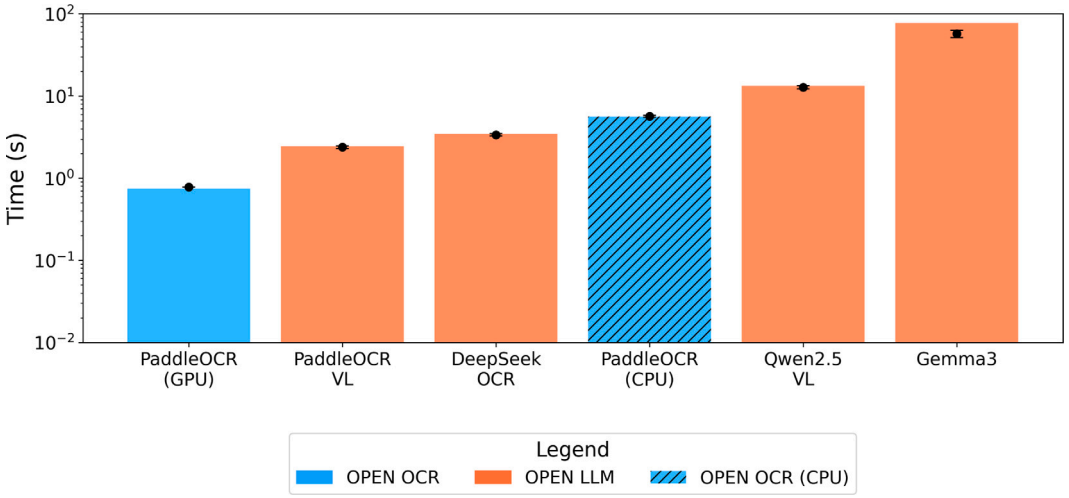

Fig. 12. Time processing metrics of open-source OCR and LLM solutions on FOX (Chinese subset).

Table 13
Aggregate costs of multimodal LLMs under token-based pricing in USD ($).

Provider	FUNSD (50)	IAM (232)	SROIE (347)	FOX (212)	Total ($)
Gemini 2.0 Flash	0.0235	0.0601	0.1250	0.1160	0.3246
Claude Haiku 4.5	0.1548	0.4701	1.0597	1.3876	3.0722
GPT-4o	0.4330	1.3622	3.6088	3.9925	9.3965

6.3. Monetary costs

Monetary cost is a crucial factor in the practical deployment of document processing systems, especially when operating at large scale. This section compares the cost structures and relative expenditures associated with the evaluated solutions. Commercial OCR services typically adopt a *page-based pricing* model, while multimodal LLMs are billed per token, with separate rates for input and output tokens. Open-source solutions, on the other hand, entail infrastructure costs that depend on the selected hardware configuration and usage regime.

All monetary values are derived from the official pricing pages of the respective providers. Detailed references and cost breakdowns for each computation are reported in the project repository.

Page-based pricing. Commercial OCR APIs exhibit highly uniform pricing, ranging between $0.001 and $0.0015 per page across major vendors such as Azure, Google, AWS, and Mistral. Mistral Document AI is included in this group as it is the only next-generation system in our benchmark offering a flat per-page billing scheme.

For the datasets considered (a total of 841 pages), the aggregate cost for page-based systems amounts to $0.84, with Azure Vision representing the most affordable option. These differences are minor at small scale but grow substantially when processing millions of pages. It should also be noted that enterprise contracts and committed volume tiers may significantly reduce per-page costs in real-world scenarios.

Token-based pricing. Multimodal LLMs such as Claude 4.5 Haiku, GPT-4o, and Gemini 2.0 Flash follow a token-based pricing model, where costs scale linearly with the number of input and output tokens processed. Each provider applies distinct rates per token, making overall cost estimation dependent on both document length and model verbosity. For each model, the total cost per page was computed accounting for all input token components, including image tokens, system prompt, and user instruction, as well as output tokens, following the pricing structure of each provider.

As shown in Table 13, Gemini 2.0 Flash consistently emerges as the most cost-efficient option, with a total cost of only $0.32 across all datasets, while GPT-4o is the most expensive at $9.39. Claude 4.5 Haiku lies in between, at around $3.07. These results highlight the strong variability induced by token pricing: although per-request costs appear low, cumulative expenses scale rapidly with large document volumes. Furthermore, token-based billing introduces variability in cost predictability, since output length depends on the model's generative behavior and prompt design.

Table 14 reports the mean number of input and output tokens per page for each model–dataset combination, where values are expressed as *input/output* tokens per page.

Table 14
Mean input/output tokens per page by model and dataset.

Model	FUNSD (50)	IAM (232)	SROIE (347)	FOX (212)
Gemini 2.0 Flash	1,303/850	2,203/97	2,152/363	1,580/973
Claude Haiku 4.5	1,041/411	1,543/97	1,316/348	1,231/1 063
GPT-4o	785/316	930/81	1,188/297	905/954

Table 15
Projected costs (USD) for processing 1000 document pages under each pricing model.

System	Pricing model	Cost per page ($)	Cost per 1k Pages ($)
Mistral Document AI	Page-based	0.0020	2.00
Azure Doc. Intel./Google Vision/Google Doc. AI/AWS Textract	Page-based	0.0015	1.50
Azure Vision	Page-based	0.0010	1.00
Gemini 2.0 Flash	Token-based	≈0.0003	≈0.38
Claude 4.5 Haiku	Token-based	≈0.0036	≈3.65
GPT-4o	Token-based	≈0.011	≈11.17
Open-source (g5.xlarge)	Infrastructure	≈0.0002	≈0.17

Cost of open-source deployment. For open-source systems, costs were estimated based solely on infrastructure usage. All experiments were executed on an AWS `g5.xlarge` instance (1 GPU A10, 4 vCPUs, 16 GB RAM), priced at approximately $1.006 per hour in on-demand mode, corresponding to around $8800 per year of continuous operation. A 1-year reserved plan reduces this cost to approximately $5600.

These costs are sustainable primarily in high-throughput scenarios. Under the infrastructure configuration and pricing assumptions adopted in this study, self-hosted open-source deployments become cost-effective compared to commercial OCR APIs above roughly six million pages per year, and competitive with efficient LLM-based solutions such as Gemini 2.0 Flash above twenty million pages per year. These break-even points should be regarded as illustrative estimates rather than universally applicable operational thresholds, as they depend on infrastructure costs, processing throughput, deployment architecture, and provider pricing policies. Although these volumes are substantial, they remain realistic for document-intensive sectors such as finance, insurance, and healthcare.

Overall, commercial OCR APIs remain the most predictable and cost-stable option for low to medium throughput, while token-based multimodal models offer superior flexibility at a higher per-page cost. Open-source systems, though initially requiring infrastructure investment, become economically advantageous at scale, especially when amortized over sustained workloads.

Projected costs at scale. To enhance transparency and replicability, Table 15 reports projected costs for processing 1000 pages under each pricing model, based on the per-page and per-token rates observed in our experiments.

For token-based systems, per-page costs were derived from the average number of input and output tokens observed across all evaluated datasets, multiplied by the official per-token rates of each provider. These figures should be interpreted as estimates, as actual costs vary with document density, image resolution, and model verbosity. For open-source systems, the per-page cost was computed as the ratio between the hourly on-demand instance cost ($1.006/h) and the observed mean throughput.

It is worth noting that several providers offer significant economies of scale that can substantially reduce per-page costs in high-volume deployments. For instance, Google Document AI applies a reduced rate of $0.60 per 1000 pages for monthly volumes exceeding 5 million pages, compared to the standard rate of $1.50 per 1000 pages. Additionally, asynchronous batch APIs offered by providers such as Mistral and Gemini can reduce token-based costs by up to 50% for workloads that do not require real-time processing. These pricing dynamics suggest that the cost landscape can shift considerably depending on deployment scale and latency requirements.[15]

6.4. Discussion

The analysis highlights the significant progress achieved by current document processing technologies and the evolving balance between OCR systems and multimodal LLMs across the three key dimensions of accuracy, processing time and monetary cost.

Accuracy and stability. Commercial OCR engines remain the most consistent and reliable option across document types, combining high recognition accuracy with low latency, and consistently exhibiting the narrowest confidence intervals on structured datasets, reflecting highly deterministic behavior across documents. Open-source OCR systems, such as *PaddleOCR* and *DocTR*, can achieve comparable performance under controlled conditions and offer additional benefits such as on-premise deployment and greater integration flexibility. However, their robustness declines in challenging scenarios, particularly with handwritten or degraded input. Recent open-source multimodal LLMs, such as *Qwen2.5-VL*, achieve state-of-the-art accuracy on both printed and handwritten

[15] All pricing figures reported in this study were retrieved from official provider documentation and are subject to change.

text, effectively matching the performance of commercial multimodal systems. This result is particularly notable considering that the evaluated configuration corresponds to the 3B-parameter variant, which represents a lightweight model within its family. Nevertheless, it is specialized multimodal solutions, such as *Mistral Document AI*, rather than general-purpose vision–language models, that are most likely to replace traditional OCR pipelines for document text transcription, as they integrate task-specific optimization with end-to-end visual–language capabilities. A recurring finding across datasets is the surprisingly strong performance of lightweight open-source LLMs, particularly *Qwen2.5-VL-3B* and the document-specialized *PaddleOCR-VL* and *DeepSeek-OCR*. On IAM and FOX English, these models match or exceed several commercial LLMs, demonstrating that competitive OCR performance is achievable without proprietary APIs. LLM-based systems, however, show wider and more variable confidence intervals, particularly on FUNSD and FOX Chinese, where instance-to-instance variability is highest. This behavioral instability has direct implications for production deployment: a system with high mean accuracy but wide intervals may produce unpredictable outputs on individual documents, requiring additional tailored validity checks or fallback mechanisms.

Printed vs handwritten documents. The contrast between IAM and the printed-text datasets is informative beyond raw accuracy figures. On HTR-style inputs, where local visual cues are inherently more ambiguous than in printed text, multimodal LLMs systematically outperform detection-based OCR engines, even open-source ones with substantial domain adaptation (PaddleOCR-FT). We interpret this as evidence that strong language priors, encoded in the decoder of vision–language models, provide a meaningful advantage when local visual evidence is under-determined, complementing rather than replacing visual recognition. Since IAM is a public dataset (as we discuss below, in Domain Adaptation), prior exposure during training may also contribute to this advantage, although the consistency of the effect across the printed and dense-text settings suggests it is not solely attributable to memorization.

Generative errors. Multimodal LLMs exhibit error categories that are structurally different from those of traditional OCR engines. The most recurrent patterns observed in our evaluation include: repetition loops, where the model replicates already-transcribed content; layout-driven hallucinations, where non-textual elements such as horizontal rules are transcribed as character sequences (e.g., `---`); omissions, where portions of dense pages are silently skipped; and boilerplate preamble insertions, where the model prepends task-related text before the actual transcription.

The role of FCA as a diagnostic tool. Across all datasets, the gap between FCA and CA proves to be one of the most informative indicators in the benchmark, revealing failure modes that CA and WA alone cannot distinguish. On SROIE and IAM, where layouts are linear and reading order is unambiguous, FCA-CA gaps are generally small and primarily reflect genuine recognition errors. On FUNSD and FOX, where form-like or multi-column layouts introduce reading order ambiguity, large FCA-CA gaps expose a systematic class of failures: systems that recognize characters correctly but serialize them in an order inconsistent with the ground truth. This pattern is most pronounced for detection-based OCR engines across all datasets, whose serialization strategy appears to systematically diverge from ground truth conventions on complex layouts. For LLM-based systems, FCA-CA gaps are generally smaller, reflecting the advantage of holistic image understanding over modular detection-based pipelines for layout-aware serialization.

Domain adaptation. Model adaptation remains a decisive factor for robust performance in real-world scenarios. Fine-tuning commercial or open-source OCR engines on domain-specific data can yield substantial improvements, particularly for non-standard fonts and layouts (e.g., handwritten text). In contrast, multimodal LLMs demonstrate stronger zero-shot generalization but currently lack lightweight fine-tuning pipelines for domain-specific adaptation, limiting their operational flexibility in enterprise settings. It should be noted, however, that the zero-shot performance of commercial multimodal LLMs on SROIE, FUNSD, and IAM may be partially influenced by training-set overlap, as these are public datasets pre-dating the training cutoff of most evaluated models, a limitation that applies to any benchmark relying on publicly available evaluation data.

Prompting strategies. CoT prompting is consistently beneficial for most LLMs on most datasets, but with some exceptions. The largest gains are observed for models with the weakest baseline performance on a given dataset, suggesting that CoT is most valuable when the task presents a non-trivial challenge for the model. Conversely, CoT provides negligible or negative effects for already-strong models on less complex datasets, and in some cases causes catastrophic degradation: *Qwen2.5-VL-3B-CoT* on FOX Chinese and *Gemma 3-4B-CoT* on FOX English both show substantial CA drops relative to their non-CoT variants, indicating that the additional step-by-step instructions can perturb output serialization on high-density inputs, for instance by inducing boilerplate preambles or omissions. These results suggest that the effects of CoT prompting cannot be assumed to generalize across models, datasets, or scripts.

Analysis of key components. Beyond aggregate performance, the results allow a cross-cutting analysis of the impact of key architectural components on OCR quality. Vision encoder design appears to be a primary driver of cross-script generalization: models equipped with high-resolution dynamic encoders, such as Qwen2.5-VL (ViT-L) and PaddleOCR-VL (NaViT), maintain competitive performance on dense text pages, whereas models relying on fixed-resolution encoders collapse on the same inputs, suggesting that resolution adaptability is critical for dense-text scenarios. Decoder specialization emerges as a stronger predictor of output reliability than raw model scale: among open-source multimodal LLMs, document-specialized decoders (PaddleOCR-VL, DeepSeek-OCR) exhibit substantially lower incidence of generative artifacts such as repetition loops, boilerplate preambles, schema drift and tighter confidence intervals, compared to general-purpose language decoders of comparable or larger size. It should be noted that this component-level analysis is necessarily limited to systems whose architecture is publicly documented. For commercial multimodal LLMs, architectural details are not disclosed, precluding analogous conclusions about the role of their internal components.

Latency. Processing time results consistently reveal a three-tier stratification across all datasets. Commercial OCR services and GPU-accelerated open-source engines occupy the fastest tier, with sub-second per-page latencies suitable for real-time applications. Document-specialized open-source LLMs such as *PaddleOCR-VL* and *DeepSeek-OCR* occupy an intermediate tier, achieving latencies of 1–3 s/page and representing an interesting middle ground between recognition quality and deployment efficiency. General-purpose multimodal LLMs are one to two orders of magnitude slower, with latency scaling directly with output length: on dense-text FOX pages, *Claude 4.5 Haiku* and *GPT-4o* reach 22–30 s/page, making them effectively unsuitable for high-throughput scenarios involving text-heavy documents. A notable exception is *Mistral Document AI*, which consistently achieves latencies comparable to commercial OCR services despite its generative architecture.

Cost. Commercial OCR services offer predictable and uniform page-based pricing, making them the most cost-efficient option at small to medium scale. Among token-based multimodal LLMs, *Gemini 2.0 Flash* is by far the most affordable, at approximately one order of magnitude cheaper than *GPT-4o* for equivalent workloads; however, token-based billing scales rapidly with document length and volume, and can become prohibitively expensive for large-scale text-heavy workloads. Open-source solutions entail no per-page cost but require infrastructure investment that becomes economically competitive with commercial OCR APIs above an estimated six million pages per year under our assumptions, and with efficient LLM-based solutions above twenty million pages per year. At large scale, volume-based pricing dynamics can substantially alter the cost landscape. Overall, commercial OCR services provide the best balance between cost, accuracy, and reliability for most deployment scenarios, while open-source deployments offer greater long-term cost efficiency and data governance for large-scale or privacy-sensitive applications.

7. Conclusions and future work

This study presented a systematic and comparative evaluation of traditional OCR systems and multimodal LLMs on real-world document text recognition tasks, analyzing their relative strengths in terms of accuracy, processing time, and monetary cost. Beyond the empirical findings, the work was conceived as a practical contribution to the document analysis and information management community, bridging research-oriented benchmarking with the operational needs of industrial applications.

The proposed evaluation framework was designed around three core dimensions — accuracy, latency, and cost — reflecting the key factors that determine the deployability of document transcription systems at scale. Unlike existing benchmarks such as OCRBench V2 (Fu et al., 2024) and OmniDocBench (Ouyang et al., 2025), which focus predominantly on accuracy-driven evaluation of multimodal LLMs, our framework provides comprehensive coverage of commercial OCR solutions alongside both proprietary and open-source multimodal LLMs, enabling a more complete picture of the current landscape. By combining printed, scanned, handwritten, and dense-text multilingual datasets, the study assessed heterogeneous conditions representative of real-world usage, while ensuring methodological consistency and reproducibility across commercial and open-source systems.

Given the fast-paced evolution of vision–language architectures, this evaluation does not seek exhaustiveness in the category of multimodal LLMs, but rather offers a reproducible and extensible framework for future comparative analyses and continuous validation of emerging OCR and multimodal models.

The results confirm commercial OCR engines remain the most mature and reliable solutions for production-grade text recognition, offering a stable balance between accuracy, scalability, and cost efficiency. Fine-tuned open-source frameworks, such as PaddleOCR, demonstrate that lightweight domain adaptation continues to be an effective strategy for achieving competitive performance in specialized document scenarios, without incurring the complexity or cost of large-scale model serving. Conversely, multimodal LLMs achieve strong performance on unstructured and handwritten inputs, highlighting their potential as a complementary component to traditional OCR systems in complex use cases, though the magnitude of this advantage on public handwriting benchmarks should be interpreted with caution given possible training-set exposure. However, their computational footprint and inference latency still limit real-time adoption, emphasizing the importance of architectural optimization and efficient deployment strategies.

These findings offer concrete guidance for researchers and practitioners in the field of document and information management. Depending on document typology, throughput requirements, and latency or cost constraints, one may choose between a consolidated OCR pipeline, ideal for structured and large-scale scenarios, or a multimodal LLM-based approach, better suited for heterogeneous or unstructured documents. Open-source solutions, while requiring greater technical expertise and initial investment, become increasingly cost-effective at scale, whereas commercial APIs remain the preferred option for predictable, low-maintenance deployments.

Taken together, the results provide not only an updated performance landscape, but also a reproducible methodological foundation to support the ongoing benchmarking, optimization, and conscious integration of emerging OCR and multimodal technologies in real-world document processing systems.

This assessment naturally suggests a number of directions for future investigation. All systems were evaluated at the single-page level; extending the evaluation to multi-page documents, including cross-page processing modalities enabled by models with extended context windows, remains an open direction. Regarding dataset coverage, while the inclusion of the FOX dataset extends our evaluation to Chinese content, coverage of additional non-Latin scripts such as Arabic or right-to-left languages would further improve the generalizability of the conclusions. A systematic evaluation of system robustness to varying input resolutions and acquisition conditions, including lower DPI scans, skewed or noisy inputs, would complement the as-is evaluation conducted in this study and provide deeper insights into the practical reliability of both OCR engines and multimodal LLMs under suboptimal document quality. A systematic exploration of prompt strategies for multimodal LLMs, beyond the fixed prompt and Chain-of-Thought condition evaluated here, could provide deeper insights into the sensitivity of these models to prompt design. Extending

the evaluation framework with semantic metrics (such as entity-level F1 scores aligned with structured dataset annotations) and assessing how output differences between systems propagate to downstream tasks such as information extraction and table parsing, would provide additional insights into the practical utility of OCR outputs beyond raw transcription accuracy. At the architectural level, a component-level analysis of multimodal LLM architectures, including the role of vision encoders and text decoders in determining OCR performance, represents a natural extension of this work. A direct measurement of energy consumption and carbon footprint of OCR pipelines would complement the cost and latency analyses with sustainability metrics, becoming increasingly relevant for large-scale document processing deployments. Finally, the complementary strengths observed between traditional OCR engines and multimodal LLMs naturally suggest the exploration of hybrid OCR-grounded architectures, in which a deterministic OCR engine provides an explicit transcription as grounding context for a multimodal LLM. Such pipelines are particularly attractive in high-consequence settings, such as legal, medical, and financial document processing, where factual correctness is critical and the generative artifacts documented in our evaluation would be unacceptable. Developing evaluation methodologies capable of quantitatively separating visual recognition contributions from language-prior contributions in multimodal LLMs would provide deeper insights into the nature of their OCR performance, complementing the qualitative error analysis introduced in this work.

CRediT authorship contribution statement

Valerio Caravani: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Data curation, Conceptualization. **Riccardo De Cesaris:** Writing – review & editing, Writing – original draft, Validation, Methodology, Conceptualization. **Paolo Merialdo:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Conceptualization.

Acknowledgments

This work was supported by myBiros S.r.l. Valerio Caravani's industrial PhD fellowship was co-funded by myBiros S.r.l. and the Lazio Region under the Public Notice issued pursuant to Decreto Dirigenziale No. G06899 of 8 June 2021 (Grant No. F83C22001250005). Riccardo De Cesaris's PhD fellowship was funded by Seeweb S.r.l. and the European Union under the NextGenerationEU programme, Mission 4, Component 2, "From Research to Business," Investment 3.3, pursuant to Italian Ministerial Decree No. 630 of 24 April 2024 (Grant No. F81J24000370004).

Data availability

Data will be made available on request.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Agazzi, O. E., & Kuo, S.-s. (1993). Hidden markov model based optical character recognition in the presence of deterministic transformations. *Pattern Recognition*, *26*(12), 1813–1826.

Anthropic (2025). Claude haiku 4.5 system card. (System card), (Accessed 25 February 2026). (2026).

Batra, P., Phalnikar, N., Kurmi, D., Tembhurne, J., Sahare, P., & Diwan, T. (2024). Ocr-mrd: performance analysis of different optical character recognition engines for medical report digitization. *International Journal of Information Technology*, *16*(1), 447–455.

Chen, X., Alayrac, J.-B., Bitton, Y., Borgeaud, S., Brock, A., Cai, T., Carlini, N., de Las Casas, D., Cherepanova, V., Chowdhery, A., et al. (2023). Pali-x: On scaling up a multilingual vision and language model. arXiv preprint arXiv:2305.18565, https://arxiv.org/abs/2305.18565.

Clausner, C., Pletschacher, S., & Antonacopoulos, A. (2020). Flexible character accuracy measure for reading-order-independent evaluation. *Pattern Recognition Letters*, *131*, 390–397.

Cui, C., Sun, T., Liang, S., Gao, T., Zhang, Z., Liu, J., et al. (2025). Paddleocr-vl: Boosting multilingual document parsing via a 0.9 b ultra-compact vision-language model. arXiv preprint arXiv:2510.14528.

Cui, C., Sun, T., Lin, M., Gao, T., Zhang, Y., Liu, J., Wang, X., Zhang, Z., Zhou, C., Liu, H., et al. (2025). Paddleocr 3.0 technical report. arXiv preprint arXiv:2507.05595.

Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Fu, L., Yang, B., Kuang, Z., Song, J., Li, Y., Zhu, L., Luo, Q., Wang, X., Lu, H., Huang, M., Li, Z., Tang, G., Shan, B., Lin, C., Liu, Q., Wu, B., Feng, H., Liu, H., Huang, C., ... Bai, X. (2024). Ocrbench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. arXiv:2501.00321, https://arxiv.org/abs/2501.00321.

G. Team (2025). Gemma 3. https://arxiv.org/abs/2503.19786.

Gabasio, A. (2013). *Comparison of optical character recognition (OCR) software*. Department of Computer Science, Faculty of Engineering, LTH, Lund University.

Gbada, H., Kalti, K., & Mahjoub, M. A. (2025). Deep learning approaches for information extraction from visually rich documents: Datasets, challenges and methods. *International Journal on Document Analysis and Recognition (IJDAR)*, *28*(1), 121–142.

Ghosh, K., Chakraborty, A., Parui, S. K., & Majumder, P. (2016). Improving information retrieval performance on ocred text in the absence of clean text ground truth. *Information Processing & Management*, *52*(5), 873–884.

Google (2025). Gemini 2.0 flash. https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-0-flash. (Accessed 13 July 2025).

Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on machine learning* (pp. 369–376). http://dx.doi.org/10.1145/1143844.1143891.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hegghammer, T. (2022). Ocr with tesseract, amazon textract, and google document ai: A benchmarking experiment. *Journal of Computational Social Science*, *5*(1), 861–882.

Huang, Z., Chen, K., He, J., Bai, X., Karatzas, D., Lu, S., & Jawahar, C. (2019). Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 international conference on document analysis and recognition* (pp. 1516–1520). IEEE.

Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al. (2024). Gpt-4o system card. arXiv preprint arXiv:2410.21276.

Jaume, G., Ekenel, H. K., & Thiran, J.-P. (2019). Funsd: A dataset for form understanding in noisy scanned documents. In *2019 international conference on document analysis and recognition workshops*: *Vol. 2*, (pp. 1–6). IEEE.

Jo, J., Koo, H. I., Soh, J. W., & Cho, N. I. (2020). Handwritten text segmentation via end-to-end learning of convolutional neural networks. *Multimedia Tools and Applications*, *79*(43), 32137–32150.

Kanai, J., Nartker, T. A., Rice, S., & Nagy, G. (1993). Performance metrics for document understanding systems. In *Proceedings of 2nd international conference on document analysis and recognition* (pp. 424–427). IEEE.

Kukich, K. (1992). Techniques for automatically correcting words in text. *ACM Computing Surveys*, *24*(4), 377–439.

Kwon, W., Shoeybi, M., Patwary, M., Zhe, S., Heinecke, A., Catanzaro, B., & Narayanan, D. (2023). Efficient memory management for large language model serving with vllm. In *Proceedings of the ACM symposium on cloud computing*. ACM, http://dx.doi.org/10.1145/3620678.3624645.

Lam-Adesina, A. M., & Jones, G. J. (2006). Examining and improving the effectiveness of relevance feedback for retrieval of scanned text documents. *Information Processing & Management*, *42*(3), 633–649.

Laurençon, H., Touvron, H., Cord, M., Jegou, H., & Douze, M. (2024). Obelics and idefics2: Scaling multimodal models with a billion images. arXiv preprint arXiv:2404.03608, https://arxiv.org/abs/2404.03608.

Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., & Wei, F. (2023). Trocr: Transformer-based optical character recognition with pre-trained models. In *Proceedings of the AAAI conference on artificial intelligence*: *Vol. 37*, (pp. 13094–13102).

Liao, H., RoyChowdhury, A., Li, W., Bansal, A., Zhang, Y., Tu, Z., Satzoda, R. K., Manmatha, R., & Mahadevan, V. (2023). Doctr: Document transformer for structured information extraction in documents. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 19584–19594).

Liao, M., Wan, Z., Yao, C., Chen, K., & Bai, X. (2020). Real-time scene text detection with differentiable binarization. In *Proceedings of the AAAI conference on artificial intelligence*: *Vol. 34*, (pp. 11474–11481).

Liu, Y., Li, Z., Huang, M., Yang, B., Yu, W., Li, C., Yin, X.-C., Liu, C.-L., Jin, L., & Bai, X. (2024). Ocrbench: On the hidden mystery of ocr in large multimodal models. *Science China. Information Sciences*, *67*(12), Article 220102.

Liu, X., Meng, G., & Pan, C. (2019). Scene text detection and recognition with advances in deep learning: a survey. *International Journal on Document Analysis and Recognition (IJDAR)*, *22*(2), 143–162.

Liu, C., Wei, H., Chen, J., Kong, L., Ge, Z., Zhu, Z., et al. (2024). Focus anywhere for fine-grained multi-page document understanding. arXiv preprint arXiv:2405.14295.

Ma, Y., Yu, D., Wu, T., & Wang, H. (2019). Paddlepaddle: An open-source deep learning platform from industrial practice. *Frontiers of Data and Computing*, *1*(1), 105–115.

Marti, U.-V., & Bunke, H. (2002). The iam-database: an english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, *5*, 39–46.

Marzal, A., & Vidal, E. (2002). Computation of normalized edit distance and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*(9), 926–932.

Mei, J., Islam, A., Moh'd, A., Wu, Y., & Milios, E. (2018). Statistical learning for ocr error correction. *Information Processing & Management*, *54*(6), 874–887.

Mindee Team (2021). doctr: document text recognition. https://github.com/mindee/doctr. (Accessed 13 July 2025).

Mistral AI (2024). Mistral ocr. https://mistral.ai/news/mistral-ocr. (Accessed 13 July 2025).

Neumann, L., & Matas, J. (2012). Real-time scene text localization and recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3538–3545). http://dx.doi.org/10.1109/CVPR.2012.6248011.

Otsu, N., et al. (1975). A threshold selection method from gray-level histograms. *Automatica*, *11*(285–296), 23–27.

Ouyang, L., Qu, Y., Zhou, H., Zhu, J., Zhang, R., Lin, Q., Wang, B., Zhao, Z., Jiang, M., Zhao, X., et al. (2025). Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. In *Proceedings of the computer vision and pattern recognition conference* (pp. 24838–24848).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd conference on neural information processing systems*.

Q. Team (2024). Qwen2 technical report. arXiv preprint arXiv:2412.15115.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *Proceedings of the 38th international conference on machine learning*: *Vol. 139*, (pp. 8748–8763). PMLR.

Rashid, S. F., Akmal, A., Adnan, M., Aslam, A. A., & Dengel, A. (2017). Table recognition in heterogeneous documents using machine learning. In *2017 14th IAPR international conference on document analysis and recognition*: *Vol. 1*, (pp. 777–782). IEEE.

Reul, C., Springmann, U., Wick, C., & Puppe, F. (2018). State of the art optical character recognition of 19th century fraktur scripts using open source engines. In *Proceedings of the international conference on frontiers in handwriting recognition* (pp. 39–44). IEEE, http://dx.doi.org/10.1109/ICFHR-2018.2018.00016.

Reynaert, M. (2008). Non-interactive ocr post-correction for giga-scale digitization projects. In *International conference on intelligent text processing and computational linguistics* (pp. 617–630). Springer.

Rice, S. V., Kanai, J., & Nartker, T. A. (1993). *An evaluation of ocr accuracy* (p. 20). Information Science Research Institute, 1993 Annual Research Report 9.

Santos, E. A. (2019). Ocr evaluation tools for the 21st century. In *Proceedings of the workshop on computational methods for endangered languages*: *Vol. 1*.

Shi, B., Bai, X., & Yao, C. (2017). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. In *IEEE transactions on pattern analysis and machine intelligence*: *Vol. 39*, (pp. 2298–2304). http://dx.doi.org/10.1109/TPAMI.2016.2646371.

Shi, Y., Peng, D., Liao, W., Lin, Z., Chen, X., Liu, C., Zhang, Y., & Jin, L. (2023). Exploring ocr capabilities of gpt-4v (ision): A quantitative and in-depth evaluation. arXiv e-prints arXiv–2310.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *3rd international conference on learning representations*. Computational and Biological Learning Society.

Singh, A., Bacchuwar, K., & Bhasin, A. (2012). A survey of ocr applications. *International Journal of Machine Learning and Computing*, *2*(3), 314.

Smith, R. (2007). An overview of the tesseract ocr engine. In *Proceedings of the ninth international conference on document analysis and recognition*: *Vol. 2*, (pp. 629–633). IEEE, http://dx.doi.org/10.1109/ICDAR.2007.4376991.

Song, M. (2026). Defining the problem: The impact of ocr quality on retrieval-augmented generation performance and strategies for improvement. *Information Processing & Management*, *63*(1), Article 104368.

Tafti, A. P., Baghaie, A., Assefi, M., Arabnia, H. R., Yu, Z., & Peissig, P. (2016). Ocr as a service: an experimental evaluation of google docs ocr, tesseract, abbyy finereader, and transym. In *International symposium on visual computing* (pp. 735–746). Springer.

Team, G., Georgiev, P., Lei, V. I., Burnell, R., Bai, L., Gulati, A., Tanzer, G., Vincent, D., Pan, Z., Wang, S., et al. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530.

van Strien, D., Beelen, K., Ardanuy, M., Hosseini, K., Mcgillivray, B., Colavizza, G., et al. (2020). Assessing the impact of ocr quality on downstream nlp tasks. In *ICAART 2020-proceedings of the 12th international conference on agents and artificial intelligence*: *Vol. 1*, (pp. 484–496). SciTePress.

Veit, A., Matera, T., Neumann, L., Matas, J., & Belongie, S. (2016). Coco-text: Dataset and benchmark for text detection and recognition in natural images. arXiv preprint arXiv:1601.07140.

Vijayarani, S., & Sakila, A. (2015). Performance comparison of ocr tools. *International Journal of UbiComp (IJU)*, *6*(3), 19–30.

Vithlani, P., & Kumbharana, C. (2015). Comparative study of character recognition tools. *International Journal of Computer Applications*, *118*(9).

Wang, X.-F., He, Z.-H., Wang, K., Wang, Y.-F., Zou, L., & Wu, Z.-Z. (2023). A survey of text detection and recognition algorithms based on deep learning technology. *Neurocomputing*, *556*, Article 126702.

Wang, A.-L., Tang, J., Lei, L., Feng, H., Liu, Q., Fei, X., Lu, J., Wang, H., Liu, W., Liu, H., et al. (2025). Wilddoc: How far are we from achieving comprehensive and robust document understanding in the wild?. arXiv preprint arXiv:2505.11015.

Wei, H., Sun, Y., & Li, Y. (2025). Deepseek-ocr: Contexts optical compression. arXiv preprint arXiv:2510.18234.

Yan, Z., Ye, Z., Ge, J., Qin, J., Liu, J., Cheng, Y., & Gurrin, C. (2025). Docextractnet: A novel framework for enhanced information extraction from business documents. *Information Processing & Management*, *62*(3), Article 104046.

Yang, Z., Tang, J., Li, Z., Wang, P., Wan, J., Zhong, H., Liu, X., Yang, M., Wang, P., Liu, Y., et al. (2024). Cc-ocr: A comprehensive and challenging ocr benchmark for evaluating large multimodal models in literacy. arXiv preprint arXiv:2412.02210.

Ye, Q., & Doermann, D. (2015). Text detection and recognition in imagery: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(7), 1480–1500. http://dx.doi.org/10.1109/TPAMI.2014.2366765.

Yin, S., Fu, C., Zhao, S., Li, K., Sun, X., Xu, T., & Chen, E. (2024). A survey on multimodal large language models. *National Science Review*, *11*(12), Article nwae403.

Zhai, X., Mustafa, B., Kolesnikov, A., & Beyer, L. (2023). Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 11975–11986).

Zhang, X., Pang, R., Wu, Y., Liu, Z., Wang, L., Zhang, H., Xu, Y., Liu, C., Awadallah, A. H., Wang, L., et al. (2023). Kosmos-2: Grounding multimodal large language models to the world. arXiv preprint arXiv:2306.14824, https://arxiv.org/abs/2306.14824.

Zhang, Z., Zhang, C., Shen, C., & Liu, Y. (2020). Deep learning for scene text detection and recognition: A survey. *Pattern Recognition*, *107*, Article 107376. http://dx.doi.org/10.1016/j.patcog.2020.107376.

Zhu, Y., Yao, C., & Bai, X. (2016). Scene text detection and recognition: Recent advances and future trends. *Frontiers of Computer Science*, *10*(1), 19–36.