

# MonkeyOCR: Document Parsing with a Structure-Recognition-Relation Triplet Paradigm

Zhang Li<sup>1</sup>, Yuliang Liu<sup>1,†</sup>, Qiang Liu<sup>2</sup>, Zhiyin Ma<sup>1</sup>, Ziyang Zhang<sup>1</sup>,  
Shuo Zhang<sup>1</sup>, Zidun Guo<sup>1</sup>, Jiarui Zhang<sup>2</sup>, Xinyu Wang<sup>1</sup>, Xiang Bai<sup>1</sup>

<sup>1</sup>Huazhong University of Science and Technology, <sup>2</sup>Kingsoft Office

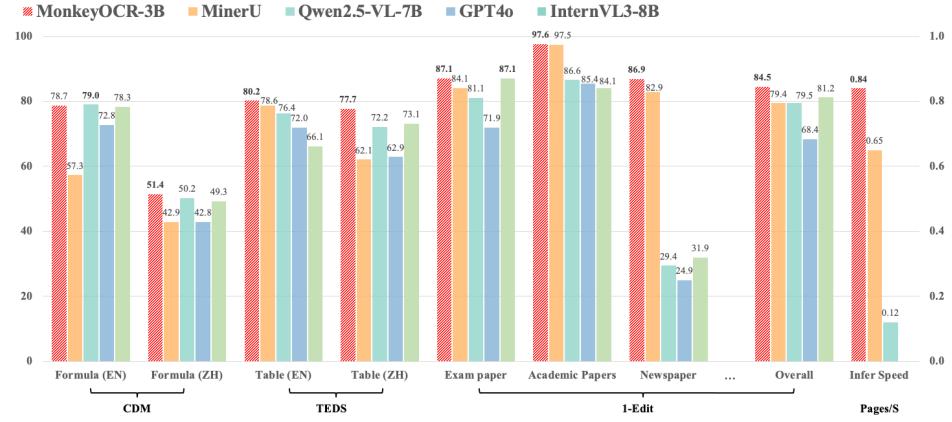


Figure 1: Performance comparison of MonkeyOCR and other SOTA models on OmniDocBench [33]. “Overall” represents the comprehensive evaluation across nine document types in OmniDocBench.

## Abstract

We introduce **MonkeyOCR**, a vision-language model for document parsing that advances the state of the art by leveraging a **Structure-Recognition-Relation (SRR)** triplet paradigm. This design simplifies what would otherwise be a complex multi-tool pipeline (as in MinerU’s modular approach) and avoids the inefficiencies of processing full pages with giant end-to-end models (e.g., large multimodal LLMs like Qwen-VL). In SRR, document parsing is abstracted into three fundamental questions – “Where is it?” (structure), “What is it?” (recognition), and “How is it organized?” (relation) – corresponding to layout analysis, content identification, and logical ordering. This focused decomposition balances accuracy and speed: it enables efficient, scalable processing without sacrificing precision. To train and evaluate this approach, we introduce the **MonkeyDoc** (the most comprehensive document parsing dataset to date), with 3.9 million instances spanning over ten document types in both Chinese and English. Experiments show that MonkeyOCR outperforms MinerU by an average of 5.1%, with particularly notable improvements on challenging content such as formulas (+15.0%) and tables (+8.6%). Remarkably, our 3B-parameter model surpasses much larger and top-performing models, including Qwen2.5-VL (72B) and Gemini 2.5 Pro, achieving state-of-the-art average performance on English document parsing tasks. In addition, MonkeyOCR processes multi-page documents significantly faster (0.84 pages per second compared to 0.65 for MinerU and 0.12 for Qwen2.5-VL-7B). The 3B model can be efficiently deployed for inference on a single NVIDIA 3090 GPU. Code and models will be released at <https://github.com/Yuliang-Liu/MonkeyOCR>.

# 1 Introduction

Document parsing is a foundational technology that transforms unstructured, multimodal content, including text, tables, images, formulas, and more, within various document formats into structured, machine-readable information. This capability underpins a wide range of real-world applications, such as automated business workflows, digital archiving, intelligent education, and medical record management, accelerating the digitization and automation of information-centric industries.

Unlike traditional OCR or basic visual recognition tasks, document parsing must contend with the diversity of layouts, layered visual hierarchies, and the seamless integration of multiple modalities. Modern documents frequently combine dense text, complex tables, mathematical expressions, embedded graphics, and handwritten annotations, often in a mix of languages and formats. This inherent complexity poses unique challenges: systems must not only detect and recognize content at a granular level, but also reconstruct the underlying structure and semantic relationships that are critical for downstream applications.

Given the multifaceted challenges posed by document parsing, existing solutions have evolved along two principal paradigms: *pipeline-based* and *end-to-end* approaches:

- **Pipeline-based approaches**, such as MinerU [43] and Marker [35], decompose the document parsing workflow into a series of specialized, fine-grained sub-tasks, including layout analysis, region segmentation, text recognition, table and formula detection, and structural reconstruction, with each step handled by distinct models or tools. This modular design enables targeted optimization of individual components and flexible integration of state-of-the-art algorithms for specific subtasks. However, a critical limitation of this paradigm is the *accumulation of errors* across the pipeline: inaccuracies at early stages, such as imperfect region detection or misclassification, propagate through subsequent modules, compounding their impact on the final output. For example, Figure 2 demonstrates how imprecise formula detection may result in partial cropping of characters from the preceding line, leading to erroneous recognition outcomes such as the inclusion of extraneous superscripts. Such cumulative errors can significantly degrade the overall reliability and accuracy of pipeline-based systems, especially when dealing with complex, information-dense documents.
- **End-to-end approaches**, exemplified by models such as Qwen2.5-VL-7B [1], seek to address the limitations of modular pipelines by processing entire document pages or substantial regions within a unified neural network. These models directly generate structured representations from raw inputs, streamlining the workflow and enabling joint optimization across tasks. Despite showing strong potential for holistic understanding, this approach still faces significant computational challenges. Modern documents often contain high-resolution, information-dense layouts, resulting in extremely long input and output sequences. The quadratic complexity of attention mechanisms and the necessity to model long-range dependencies impose substantial limitations on both *inference speed* and model scalability, particularly in large-scale or production environments. For example, the inference speed of Qwen2.5-VL-7B [1] is merely 18% of MinerU [43] and performs worse overall, as illustrated in Figure 1.

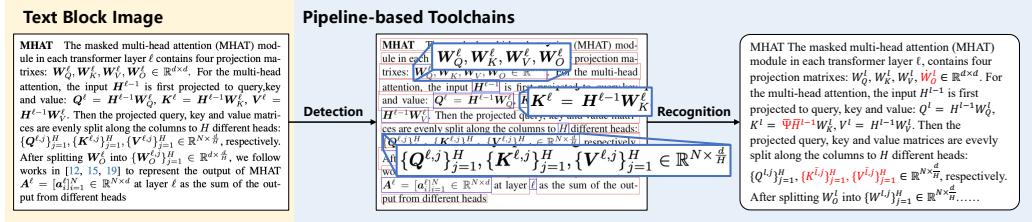


Figure 2: **Illustration of error propagation in pipeline-based document parsing toolchains.** In the middle panel, the pipeline’s detection module inaccurately segments a formula region, causing part of the formula to overlap with text from the preceding line. This leads to recognition errors in the right panel, where extraneous superscript characters are mistakenly included in the formula output.

To address the limitations of existing document parsing approaches and achieve an optimal balance between accuracy and efficiency, we introduce **MonkeyOCR**, a novel system based on a **Structure-Recognition-Relation (SRR)** triplet paradigm. In contrast to previous approaches, MonkeyOCR

SRR design:

1. structure detection/location - detects where extraction needs to be done (segment regions)
2. each region - classified into category (table/formula)
3. detecting relation b/w regions

first performs **block-level structure** detection to accurately segment semantic regions, including text blocks, tables, formulas, images, and other components within each document. Each region is then subjected to **recognition** by a unified, Large Multimodal Model (LMM), enabling end-to-end recognition across diverse content types without the error propagation typical of traditional pipelines. Finally, a block-level reading order prediction mechanism models the **relations** between detected regions, reconstructing their logical and semantic connections to generate high-fidelity structured outputs. This SRR design effectively combines the interpretability and modularity of pipeline methods with the global optimization and simplicity of end-to-end architectures.

To support robust and generalizable model training, we further develop **MonkeyDoc**, the largest and most diverse document parsing dataset to date. MonkeyDoc comprises **3.9 million** block-level instances, covering five core document parsing tasks and over ten document types, with full support for both Chinese and English. The dataset is constructed through a multi-stage pipeline that integrates meticulous manual annotation, programmatic synthesis, and model-driven automatic labeling, ensuring both quality and coverage.

Extensive experiments demonstrate that MonkeyOCR achieves state-of-the-art overall performance across nine document types, including academic papers, textbooks, handwritten notes, and densely formatted newspapers, in both Chinese and English. Compared to the leading pipeline-based system MinerU [43] and the advanced end-to-end model Qwen2.5-VL-7B [1], MonkeyOCR delivers substantial improvements: **8.6% higher TEDS** for table recognition, **15.0% better CDM** for formula recognition, and over **10% lower edit distance** on both English and Chinese tasks. Notably, our 3B model even outperforms the much larger Qwen2.5-VL-72B [1] and Google’s flagship commercial model, Gemini 2.5 Pro, on key English benchmarks. In terms of efficiency, MonkeyOCR achieves an inference speed of 0.24 pages per second for single-page documents (compared to MinerU’s 0.28 pages per second) and 0.84 pages per second for multi-page documents (compared to MinerU’s 0.65 and 0.12 for Qwen2.5-VL-7B pages per second), offering competitive or superior throughput in real-world scenarios.

## 2 Related Work

Recent advances in document parsing have shifted the field from traditional rule-based and heuristic methods to deep learning-based techniques, enabling significant improvements in robustness and adaptability across diverse document formats. Contemporary approaches can be broadly classified into two categories: **pipeline-based** and **end-to-end** approaches, each offering distinct trade-offs in flexibility, scalability, and performance.

### 2.1 Pipeline-based Approaches

Pipeline-based approaches [4; 24; 35; 43] decompose the document parsing workflow into a sequence of specialized sub-tasks, such as layout analysis [48; 14; 41], reading order prediction [44], Optical Character Recognition (OCR) [15; 17], formula recognition [42], and table structure recognition [34; 12; 46], with each task handled by a dedicated model or module. This modular design enables independent optimization and straightforward integration of advanced algorithms at each stage.

Representative works include Docling [24], which implements a linear pipeline to extract content from PDF files and generate structured outputs in JSON or Markdown formats, and Marker [35], which supports a broad range of document types (e.g., images, PDFs, PPTX, DOCX) and exports to various formats via a combination of Surya OCR, layout analysis, reading order prediction, and table recognition modules. Marker [35] further integrates LLM-based components to enhance cross-page table merging and inline mathematical expression parsing. MinerU [43] extends this paradigm with a modular architecture for layout detection, content recognition (including text, formulas, and tables), and output structuring.

Although pipeline-based approaches have driven much progress through modular design and flexible integration of specialized models, their tendency to propagate errors across sequential stages remains a key bottleneck, particularly for complex, high-density documents. The proposed MonkeyOCR seeks a more robust alternative that decouples error sources and enables efficient, high-fidelity parsing across diverse document scenarios.

## 2.2 End-to-end Approaches

End-to-end approaches aim to simplify document parsing by directly processing entire document pages or substantial regions through a unified neural network, generating structured outputs in a single stage without the need for multiple specialized models. Early representative methods include **Donut** [16], which introduces an OCR-free, Transformer-based framework unifying tasks such as document classification, visual question answering, and information extraction; and **Nougat** [3], which focuses on converting document images into well-formatted markup text. **GOT** [45] generalizes OCR to recognize diverse optical signals, including text, formulas, tables, charts, musical scores, and geometric shapes, under a unified *character* concept. The **SPTS** series [36; 23] and **OmniParser** [40] further advance unified text detection and recognition, as well as key information extraction and table parsing in both natural scene and document images.

Recent progress has been driven by Large Multimodal Models (LMMs) trained on massive document corpora. For example, **Monkey** [19; 22] enhances document understanding through high-resolution image cropping strategies, while **mPLUG-DocOwl2** [13] introduces cross-page modeling for structural reasoning in multi-page documents. **InternVL3** [5] leverages joint pretraining on text and multimodal data to enable improved cross-modal alignment and long-context understanding. **Qwen2.5-VL** [1] proposes a fine-grained parsing format (QwenVL HTML) that captures both textual content and layout, supporting accurate reconstruction of diverse document types. Meanwhile, **olmOCR** [38] is tailored for document parsing through large-scale PDF corpus finetuning and efficient inference via the **SGLang** [50] framework.

While end-to-end approaches have advanced the field by enabling unified modeling and reducing manual intervention, practical deployment in real-world document parsing remains constrained by scalability and efficiency concerns, particularly when handling heterogeneous, large-scale, or multilingual collections. Addressing these persistent gaps motivates our work, which seeks a more effective balance between recognition accuracy and computational efficiency for robust document understanding across diverse scenarios.

## 2.3 Document Parsing Dataset

A variety of datasets have been developed to support fine-grained document parsing tasks, including layout detection, reading order prediction, table and formula recognition, text extraction, and code block identification, across a wide spectrum of document types such as textbooks [6], academic papers [46; 37; 53], newspapers [6; 8], and slides [8].

For general layout analysis, **M<sup>6</sup>Doc** [6] provides 9,080 diverse document images labeled with 74 layout categories, while **CDLA** [18] focuses on Chinese academic papers, and **D4LA** [8], derived from **RDL-CDIP** [11], offers annotated data for 12 document types with rich layouts. **DocLayNet** [37] further expands this coverage, containing over 80,000 pages of human-annotated layout segmentation data from various sources.

Datasets for specific element recognition are also well-represented: **FinTabNet** [51] provides detailed structural annotations of complex tables from financial reports; **PubTabNet** [52] includes over 560,000 table images from scientific literature, each with a corresponding HTML representation; **Unimer-1M** [42] offers more than one million complex mathematical expression instances; and **HME-100k** [47] is a large-scale real-world handwritten mathematical expression recognition dataset. For reading order prediction, **ReadingBank** [44] is a weakly supervised benchmark containing 500,000 diverse document images with word-level reading order annotations. However, its focus on word-level annotations presents challenges for directly evaluating block-level reading order, which is critical for complex layouts such as multi-column or cross-page documents [20]. Additionally, **DocGenome** [46] is a large-scale structured document dataset created by annotating 500,000 scientific papers from the arXiv open-access repository, spanning 153 disciplines and covering all major document parsing tasks, making it a comprehensive English-language academic corpus.

Despite the diversity and scale of existing datasets, most are limited to single tasks, specific document types, or a single language, and few offer large-scale, fine-grained annotations across both English and Chinese or multiple document domains. This gap continues to motivate the development of more comprehensive resources for robust and versatile document parsing.

### 3 MonkeyDoc Dataset

Existing document parsing datasets typically focus on single tasks, specific document types, or a single language, which limits their effectiveness for developing and evaluating comprehensive document parsing systems. To address the need for a more versatile and large-scale resource that is capable of supporting end-to-end document parsing across a wide variety of real-world scenarios, we introduce **MonkeyDoc**.

MonkeyDoc is designed to cover a complete range of document parsing tasks, including layout detection, reading order prediction, text recognition, table recognition, formula recognition, and code block recognition. As shown in Table 1, MonkeyDoc spans more than ten diverse document domains and provides high-quality annotations in both Chinese and English, making it the most comprehensive resource of its kind to date. In comparison, previous datasets are often limited to a subset of tasks, a single document type, or monolingual settings. MonkeyDoc uniquely enables multi-task, multi-domain, and bilingual training and evaluation, supporting both fine-grained and holistic document understanding.

To achieve this breadth and depth, we developed an integrated data generation pipeline (see Figure 4) that combines filtering and harmonization of existing public datasets, meticulous manual annotation, programmatic data synthesis, and expert model-driven automatic labeling. This pipeline is organized around three core stages: **Structure Detection**, **Content Recognition**, and **Relation Prediction**. For each stage, we leverage a combination of open-source resources, advanced annotation strategies, and model-assisted workflows, ensuring high annotation quality, data diversity, and scalability across languages and document types. This multi-faceted design makes MonkeyDoc a foundational resource for robust model training, benchmarking, and deployment in both academic and industrial applications. In the following sections, we provide a detailed description of the data construction process for each stage.

Dataset	Document Domain	Supporting Tasks					Language	
		Layout Detection	Reading Order Prediction	Formula Recognition	Table Recognition	Text Recognition	EN	ZH
<i>Layout Detection Dataset</i>								
M <sup>6</sup> Doc [6]	6	✓					✓	✓
CDLA [18]	1	✓						✓
D4LA [8]	5	✓					✓	
DocLayNet [37]	2	✓					✓	✓
<i>Formula Recognition Dataset</i>								
Unimernet [42]	-			✓				
TexTeller [31]	-			✓				
<i>Table Recognition Dataset</i>								
FinTabNet [51]	-				✓			
PubTabNet [52]	-				✓			
<i>Comprehensive Dataset</i>								
DocGenome [46]	1	✓	✓	✓	✓	✓	✓	
<b>MonkeyDoc</b>	<b>&gt;10</b>	<b>✓</b>	<b>✓</b>	<b>✓</b>	<b>✓</b>	<b>✓</b>	<b>✓</b>	<b>✓</b>

Table 1: Comparison of MonkeyDoc with Other Document Parsing Datasets. MonkeyDoc covers all document parsing tasks and the largest variety of document types, in both Chinese and English.

#### 3.1 Structure Detection

Structure detection aims to identify and localize key elements within documents, such as text blocks, tables, images, and other layout components, by assigning category labels and bounding box coordinates to each region. Constructing a high-quality structure detection dataset poses several challenges, including inconsistent annotation standards across public datasets and the scarcity of annotated Chinese documents.

To address these issues, we begin by aggregating and filtering data from several publicly available structure detection datasets, including M<sup>6</sup>Doc [6], DocLayNet [37], D4LA [8], and CDLA [18],

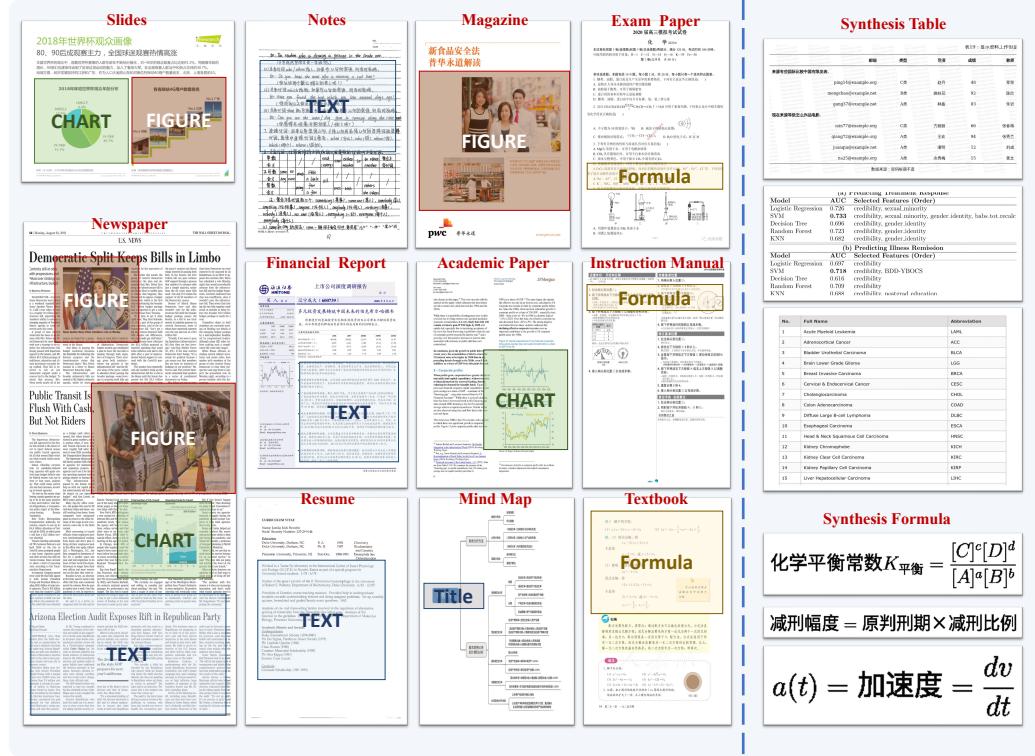


Figure 3: **Visualization of the MonkeyDoc dataset.** MonkeyDoc encompasses more than ten document types and includes our synthesized tables and formulas data.

covering both Chinese and English documents, with a total of 88k pages. We harmonize category labels by mapping disparate annotation schemes to a unified set of eleven classes, following the conventions established in MinerU [43]. To ensure consistency and data quality, we remove nested bounding boxes by retaining only the largest region for each element and filter out low-information instances by discarding bounding boxes whose area is less than 35% of the page. These steps collectively improve both the reliability and comparability of the annotations.

Given the limited availability of high-quality Chinese data, we supplement our dataset through targeted synthesis. We collect over 300,000 pages spanning more than ten types of Chinese documents (see Figure 3), such as financial reports, textbooks, and academic papers. After initial pre-annotation with existing structure detection models, we conduct post-processing, including nested box removal and low-information filtering, to obtain 28,000 high-quality samples. Additionally, we manually verified and refined a subset of pre-annotated data, resulting in a further 13,000 manually corrected Chinese samples. Through this multi-source, multi-step process, we construct a diverse and consistent structure detection dataset that supports robust model training and evaluation across both English and Chinese document types.

### 3.2 Content Recognition

Content recognition encompasses the identification and transcription of essential document elements, including text blocks, tables, formulas, and code blocks, across diverse formats and languages. The content recognition pipeline integrates several complementary strategies to maximize coverage and annotation quality:

**Cropping Document Elements.** Based on the layout annotations produced in the *Structure Selection* stage, we segment and crop individual elements, including text blocks, formula regions, tables, and code blocks, from the original document images, resulting in a total of 1.9 million samples. Partial elements are transcribed and labeled using Google’s flagship commercial model (Gemini 2.5 Pro) to ensure annotation accuracy.

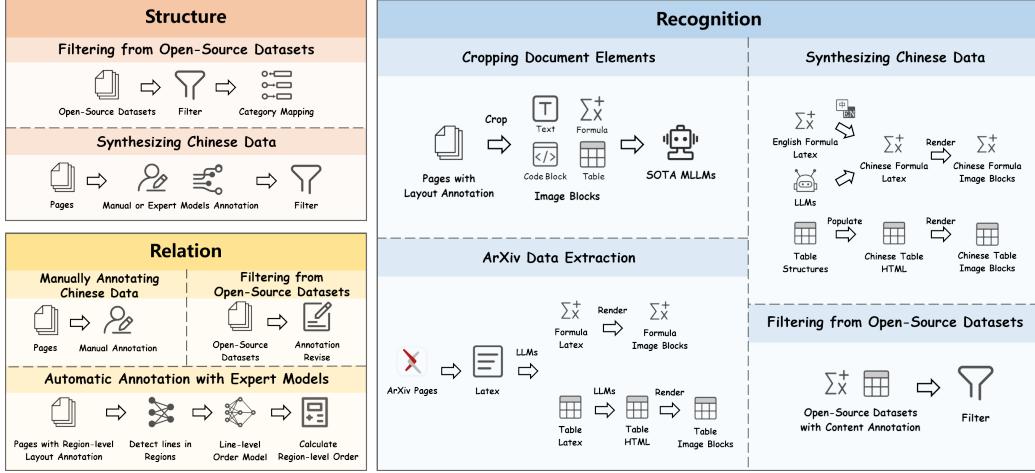


Figure 4: **Overview of the three core stages in the MonkeyDoc data generation pipeline.** **Structure Detection** aggregates and harmonizes open-source datasets, supplemented with synthesized high-quality Chinese samples; **Content Recognition** utilizes both manual and automated annotation, including synthetic data generation and element extraction; **Relation Prediction** combines manual annotation with model-assisted strategies to establish reading order and region relationships.

**Filtering from Open-Source Datasets.** For table recognition, we select and refine data from PubTabNet [52], applying stringent quality checks such as HTML tag closure, header presence, merged cell validation, header-body alignment, abnormal character detection, and syntax verification. This process yields a curated dataset of 470,000 tables. For formula recognition, we leverage the UniMER-1M [42] dataset, which aggregates formulas from diverse public sources, including Pix2tex [2], CROHME [28; 29; 25] and HME100K [47], and large-scale collections of LaTeX expressions from ArXiv, Wikipedia, and StackExchange, covering both printed and handwritten styles.

**Synthesizing Chinese Data.** To mitigate the shortage of Chinese samples for table and formula recognition, we programmatically synthesize data with high structural diversity. For tables, we construct layouts with various row and column configurations, populate them with Chinese content, and render paired HTML and image data. For formulas, we generate commonly used expressions in Chinese with large multimodal models, and also translate and render English formulas from UniMER-1M [42] using large language models. This approach produces a total of 526,000 additional Chinese samples.

**arXiv Data Extraction.** To further enhance the dataset’s breadth, we extract and process LaTeX source data for tables and formulas from arXiv papers. Irrelevant content is filtered using large language models, and the resulting data is rendered into images and structured annotations, adding 36,000 high-quality academic samples.

### 3.3 Relation Prediction

Relation prediction focuses on determining the logical reading order among detected document elements, which is essential for reconstructing coherent and semantically faithful document content, especially in cases involving complex layouts, multi-column pages, or cross-page structures. Building a region-level reading order dataset presents significant challenges due to the scarcity of annotated data and the complexity of diverse document forms.

**Open-source Data Refinement.** The primary open-source resource is DocGenome [46], which provides region-level reading order annotations generated through automated labeling. However, these annotations can be noisy or incomplete, particularly for elements like images and tables. To improve annotation accuracy, we refine these labels by explicitly associating each image and table with its corresponding caption, and filter out low-quality samples, such as pages with unannotated regions or excessive blank areas. We further score each page based on the diversity of element types,

selecting high-scoring pages for inclusion. This results in a curated set of 951,000 high-quality samples.

**Manual Annotation for Chinese Documents.** To address the limited availability of Chinese region-level reading order annotations, we manually annotate a diverse set of Chinese documents, including research reports, academic papers, user manuals, books, test papers, slides, official documents, newspapers, journals, and contracts. This effort produces an additional 154,000 high-quality samples, substantially enhancing the representation of Chinese document scenarios.

**Expert Model-Based Auto-Annotation.** For datasets that provide only region-level bounding boxes without reading order information, we leverage expert models to generate region-level reading order annotations automatically. Specifically, we utilize PPOCR [17] for line-wise text recognition within each region, obtain text line positions, and then apply LayoutReader [44] to predict the reading order of these lines. The region-level order is determined by aggregating the predicted order of all text lines within each region. Through this approach, we generate 78,000 additional region-level annotations, further enriching the diversity and coverage of our dataset.

## 4 MonkeyOCR

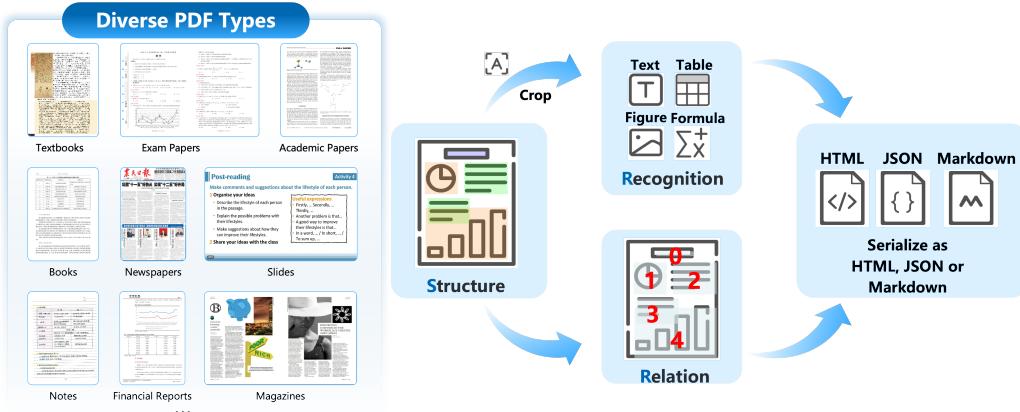


Figure 5: **The overall architecture of MonkeyOCR.** The system adopts a Structure-Recognition-Relation framework, consisting of structure detection, which locates and classifies semantic regions; block-level content recognition, which extracts structured information from each region in parallel; and relation prediction, which determines the logical reading order of the detected elements.

The proposed method, **MonkeyOCR**, addresses the fundamental limitations of both pipeline-based and end-to-end document parsing approaches by introducing a modular yet globally optimized Structure-Recognition-Relation (SRR) framework. As illustrated in Figure 5, we decompose the document parsing process into three relatively independent but tightly integrated stages: *structure detection*, *block-level content recognition*, and *relation prediction*. This design aims to mitigate the cumulative error typically observed in pipeline toolchains, while also improving inference efficiency by reducing the context length compared to monolithic end-to-end models.

In the first stage, a YOLO-based [49] document layout detector processes the input image  $I \in \mathbb{R}^{H \times W \times 3}$ , producing a set of bounding boxes  $B = \{b_1, b_2, \dots, b_n\}$  and their corresponding element types  $T = \{t_1, t_2, \dots, t_n\}$ . Each bounding box  $b_i = (x_{1i}, y_{1i}, x_{2i}, y_{2i})$  represents the spatial coordinates of the  $i$ -th element, and the element type  $t_i \in \{\text{text}, \text{table}, \text{formula}, \text{figure}, \dots\}$  specifies the category of the detected element.

For the second stage, we perform block-level content recognition in parallel. Each detected region  $b_i$  is cropped and, together with a type-specific prompt  $p_{t_i}$ , is fed into our LMM for type-aware content extraction:

$$C = \text{LMM}(\{I_{\text{crop}}^1, I_{\text{crop}}^2, \dots, I_{\text{crop}}^n\}, \{p_{t_1}, p_{t_2}, \dots, p_{t_n}\}),$$

where  $I_{\text{crop}}^i$  denotes the region cropped based on the bounding box  $b_i$ , and  $C = \{c_1, c_2, \dots, c_n\}$  denotes the structured content outputs.

In the final stage, relation prediction is carried out to infer the logical reading order among detected elements. The set of bounding boxes  $B = \{b_1, b_2, \dots, b_n\}$  input to a dedicated block-level reading order model, which predicts a sequence  $S = \{s_1, s_2, \dots, s_n\}$ , assigning each element a position in the final reading sequence. The recognized content is then reassembled as  $D = \{c_{s_1}, c_{s_2}, \dots, c_{s_n}\}$ .

This SRR pipeline ensures accurate structural understanding, precise region-level recognition, and faithful logical ordering, all while achieving improved efficiency and robustness compared to prior approaches. By modularizing the parsing process and leveraging block-wise parallelism, MonkeyOCR provides a scalable and reliable solution for real-world document parsing scenarios.

## 5 Experiments

To validate the effectiveness of MonkeyOCR, we conducted a comprehensive comparison with both open-source and closed-source methods on OmniDocBench [33]. OmniDocBench is a benchmark designed to evaluate real-world document parsing capabilities. It comprises 981 PDF pages spanning 9 document types, 4 layout styles, and 3 language categories. Through this benchmark, we are able to perform a thorough assessment of MonkeyOCR’s document parsing capabilities.

Model Type	Methods	Overall <sup>Edit ↓</sup>		Text <sup>Edit ↓</sup>		Formula <sup>Edit ↓</sup>		Formula <sup>CDM ↑</sup>		Table <sup>TEDS ↑</sup>		Table <sup>Edit ↓</sup>		Read Order <sup>Edit ↓</sup>	
		EN	ZH	EN	ZH	EN	ZH	EN	ZH	EN	ZH	EN	ZH	EN	ZH
Pipeline Tools	MinerU [43]	0.150	0.357	0.061	0.215	0.278	0.577	57.3	42.9	78.6	62.1	0.180	0.344	<b>0.79</b>	0.292
	Marker [35]	0.336	0.556	0.080	0.315	0.530	0.883	17.6	11.7	67.6	49.2	0.619	0.685	0.114	0.340
	Mathpix [26]	<b>0.191</b>	<b>0.365</b>	<b>0.105</b>	<b>0.384</b>	<b>0.306</b>	<b>0.454</b>	<b>62.7</b>	<b>62.1</b>	<b>77.0</b>	<b>67.1</b>	<b>0.243</b>	<b>0.320</b>	<b>0.108</b>	<b>0.304</b>
	Docing [24]	0.589	0.909	0.416	0.987	0.999	1	-	-	61.3	25.0	0.627	0.810	0.313	0.837
	Pix2Text [4]	0.320	0.528	0.138	0.356	0.276	0.611	78.4	39.6	73.6	66.2	0.584	0.645	0.281	0.499
	Unstructured [39]	0.586	0.716	0.198	0.481	0.999	1	-	-	0	0.06	1	0.998	0.145	0.387
Expert VLMs	OpenParse [9]	0.646	0.814	0.681	0.974	0.996	1	0.11	0	64.8	27.5	0.284	0.639	0.595	0.641
	GOT-OCR [45]	0.287	0.411	0.189	0.315	0.360	0.528	74.3	45.3	53.2	47.2	0.459	0.520	0.141	0.280
	Nougat [3]	0.452	0.973	0.365	0.998	0.488	0.941	15.1	16.8	39.9	0	0.572	1.000	0.382	0.954
	Mistral OCR [27]	0.268	0.439	0.072	0.325	0.318	0.495	64.6	45.9	75.8	63.6	0.600	0.650	0.083	0.284
	OLMOCR-sglang [38]	0.326	0.469	0.097	0.293	0.455	0.655	74.3	43.2	68.1	61.3	0.608	0.652	0.145	0.277
	SmolDocing-256M [30]	0.493	0.816	0.262	0.838	0.753	0.997	32.1	0.55	44.9	16.5	0.729	0.907	0.227	0.522
General VLMs	GPT4o [32]	0.233	0.399	0.144	0.409	0.425	0.606	72.8	42.8	72.0	62.9	0.234	0.329	0.128	0.251
	Qwen2.5-VL-7B [1]	0.312	0.406	0.157	0.228	0.351	0.574	<b>79.0</b>	50.2	76.4	72.2	0.588	0.619	0.149	0.203
	InternVL3-8B [5]	0.314	0.383	0.134	0.218	0.417	0.563	78.3	49.3	66.1	73.1	0.586	0.564	0.118	0.186
Mix	<b>MonkeyOCR-3B</b>	<b>0.140</b>	<b>0.297</b>	<b>0.058</b>	<b>0.185</b>	<b>0.238</b>	<b>0.506</b>	<b>78.7</b>	<b>51.4</b>	<b>80.2</b>	<b>77.7</b>	<b>0.170</b>	<b>0.253</b>	<b>0.093</b>	<b>0.244</b>
	MonkeyOCR-3B*	0.154	0.277	0.073	0.134	0.255	0.529	78.5	50.8	78.2	76.2	0.182	0.262	0.105	<b>0.183</b>

Table 2: The end-to-end evaluation results of different tasks on OmniDocBench. \* represents the use of the layout model trained by us with improved capability for Chinese layout detection.

### 5.1 Comparison with Other Methods on Different Tasks

Document parsing encompasses a variety of sub-tasks, including text recognition, formula recognition, table recognition, reading order detection, and more. To evaluate MonkeyOCR’s performance across these tasks, we compared it with several widely-used methods on OmniDocBench [33], including pipeline tools [43; 35], expert VLMs [45; 27], closed-source general VLMs [32], and open-source general VLMs [5; 1]. As shown in Table 2, MonkeyOCR achieves the best overall performance on both Chinese and English document parsing tasks. In particular, MonkeyOCR surpasses MinerU [43] by over 6% in overall edit distance for Chinese documents, exceeds MinerU by an average of 15.0% in formula recognition across Chinese and English, and outperforms MinerU by 8.6% on average in table recognition for both languages. Compared to Mistral OCR [27], MonkeyOCR improves average overall edit distance by 13.8% for Chinese and English, with gains of 9.8% in formula recognition and 9.3% in table recognition. Additionally, we trained a specialized version of MonkeyOCR for Chinese documents - MonkeyOCR\*. On OmniDocBench, MonkeyOCR\* achieves state-of-the-art performance in Chinese document parsing, surpassing the original MonkeyOCR by 2%.

### 5.2 Comparison with Other Methods Across Document Types

To further evaluate MonkeyOCR’s capability in handling diverse document types, we conducted a comprehensive comparison on the OmniDocBench [33] benchmark across nine categories of documents. As shown in Table 3, MonkeyOCR achieved the best overall performance across all nine

Model Type	Models	Book	Slides	Financial Report	Textbook	Exam Paper	Magazine	Academic Papers	Notes	Newspaper	Overall
Pipeline Tools	MinerU [43]	0.055	0.124	0.033	0.102	0.159	<b>0.072</b>	0.025	0.984	0.171	0.206
	Marker [35]	0.074	0.340	0.089	0.319	0.452	0.153	0.059	0.651	0.192	0.274
	Mathpix [26]	0.131	0.220	0.202	0.216	0.278	0.147	0.091	0.634	0.690	0.300
Expert VLMs	GOT-OCR [45]	0.111	0.222	0.067	0.132	0.204	0.198	0.179	0.388	0.771	0.267
	Nougat [3]	0.734	0.958	1.000	0.820	0.930	0.830	0.214	0.991	0.871	0.806
General VLMs	GPT4o [32]	0.157	0.163	0.348	0.187	0.281	0.173	0.146	0.607	0.751	0.316
	Qwen2.5-VL-7B [1]	0.148	<b>0.053</b>	0.111	0.137	0.189	0.117	0.134	0.204	0.706	0.205
	InternVL3-8B [5]	0.163	0.056	0.107	0.109	<b>0.129</b>	0.100	0.159	<b>0.150</b>	0.681	0.188
Mix	<b>MonkeyOCR-3B</b>	<b>0.046</b>	<b>0.120</b>	<b>0.024</b>	<b>0.100</b>	<b>0.129</b>	<b>0.086</b>	<b>0.024</b>	<b>0.643</b>	<b>0.131</b>	<b>0.155</b>
	MonkeyOCR-3B*	0.054	0.203	0.038	0.112	0.138	0.111	0.032	0.194	0.136	0.120

Table 3: The end-to-end text recognition performance on OmniDocBench across 9 PDF page types. \* represents the use of the layout model trained by us with improved capability for Chinese layout detection.

types. Specifically, it attained the highest end-to-end recognition accuracy in six categories. The 3B model outperformed InternVL3-8B [5] by 5% and surpassed MinerU [43] by 3.3% in overall accuracy. Notably, on the newspaper category, MonkeyOCR outperformed the previous state-of-the-art MinerU by 4%, demonstrating its strong capability in parsing dense and complex layouts. These results highlight MonkeyOCR’s superior generalization ability and robustness across various document types. Moreover, benefiting from enhanced Chinese language capabilities, MonkeyOCR\* outperforms the original version by 44.9% on the notes category, achieving state-of-the-art overall performance.

### 5.3 Implement Details

During the training process, we utilize the AdamW optimizer with a learning rate of 2e-5 and a cosine learning rate schedule. We employ a batch size of 64. Our 3B model was trained for 53 hours on 32 A800 GPUs. By integrating with LMDeploy [7], our model can successfully run on RTX 3090 GPUs.

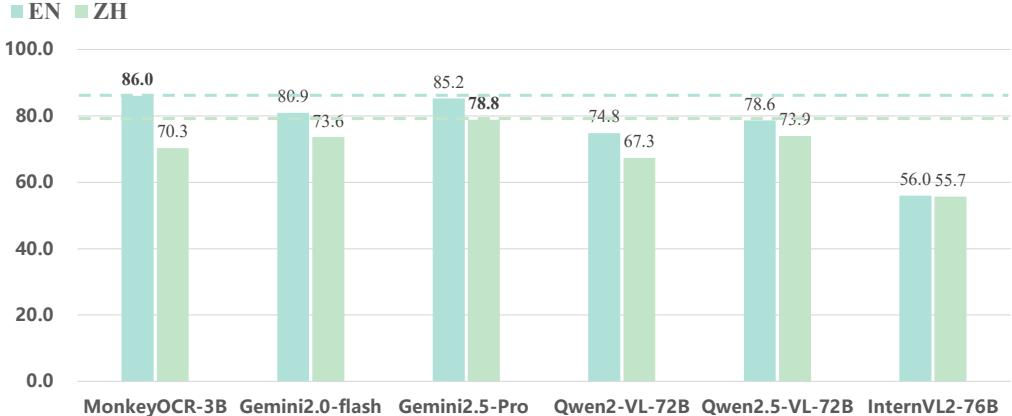


Figure 6: **End-to-end evaluation on OmniDocBench.** Performance comparison of MonkeyOCR with closed-source and extra-large open-source VLMs across different document parsing tasks.

## 6 Discussion

As is well-established, increasing model scale generally leads to improved performance. To further explore the potential of MonkeyOCR, we conducted comparative evaluations against both larger open-source models and leading closed-source commercial solutions on OmniDocBench. As illustrated in Figure 6, **MonkeyOCR achieves the highest overall performance on English documents**, outperforming Qwen2.5-VL-72B by **7.4%** and surpassing the current state-of-the-art closed-source model, Gemini 2.5-Pro, by **0.8%**. However, Gemini 2.5-Pro demonstrates slightly better performance on Chinese documents, indicating there is still some margin for improvement in MonkeyOCR’s Chinese document parsing capabilities.

## 7 Conclusion

In this work, we present MonkeyOCR, a document parsing model built on the Structure-Recognition-Relation (SRR) triplet paradigm, which unifies structural detection, content recognition, and relational ordering into a streamlined framework. This design simplifies traditional multi-tool pipelines while avoiding the inefficiencies of directly processing entire pages with LMMs, enabling both high accuracy and efficient deployment. Supported by MonkeyDoc—a large-scale, diverse dataset spanning a wide range of document types in Chinese and English—MonkeyOCR achieves strong results across benchmarks, outperforming leading open-source (e.g., MinerU [43] and Qwen2.5-VL [1]) and even closed-source models (e.g., Gemini2.5-Pro [10]) in English document parsing. Beyond its immediate performance, MonkeyOCR has the potential to serve as a foundation model for the text domain, enabling unified understanding and reasoning over complex document structures [21].

## References

- [1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- [2] Lukas Blecher. pix2tex - latex ocr. <https://github.com/lukas-blecher/LaTeX-OCR>, 2022. Accessed: 2024-02-29.
- [3] Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. Nougat: Neural optical understanding for academic documents. *arXiv preprint arXiv:2308.13418*, 2023.
- [4] breezedeus. Pix2text: An open-source python3 tool for recognizing layouts, tables, math formulas (latex), and text in images. <https://github.com/breezedeus/pix2text>, 2025.
- [5] Zhe Chen, Jiannan Wu, Wenhui Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024.
- [6] Hiuyi Cheng, Peirong Zhang, Sihang Wu, Jiaxin Zhang, Qiyuan Zhu, Zecheng Xie, Jing Li, Kai Ding, and Lianwen Jin. M6doc: a large-scale multi-format, multi-type, multi-layout, multi-language, multi-annotation category dataset for modern document layout analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15138–15147, 2023.
- [7] LMDeploy Contributors. Lmdeploy: A toolkit for compressing, deploying, and serving llm. <https://github.com/InternLM/lmdeploy>, 2023.
- [8] Cheng Da, Chuwei Luo, Qi Zheng, and Cong Yao. Vision grid transformer for document layout analysis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 19462–19472, 2023.
- [9] Sergey Filimonov. Open parse: Visually-driven document chunking for llm applications. <https://github.com/Filimoa/open-parse>, 2025.
- [10] Google DeepMind. Gemini 2.5. <https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/>, 2025.
- [11] Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *2015 13th international conference on document analysis and recognition (ICDAR)*, pages 991–995. IEEE, 2015.
- [12] Yelin He, Xianbiao Qi, Jiaquan Ye, Peng Gao, Yihao Chen, Bingcong Li, Xin Tang, and Rong Xiao. Pingan-vcgroup’s solution for icdar 2021 competition on scientific table image recognition to latex. *ArXiv*, abs/2105.01846, 2021.

- [13] Anwen Hu, Haiyang Xu, Liang Zhang, Jiabo Ye, Ming Yan, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. mplug-docowl2: High-resolution compressing for ocr-free multi-page document understanding. *arXiv preprint arXiv:2409.03420*, 2024.
- [14] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM international conference on multimedia*, pages 4083–4091, 2022.
- [15] Jaided AI. Easyocr: Ready-to-use ocr with 80+ supported languages. <https://github.com/JairedAI/EasyOCR>, 2024.
- [16] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *European Conference on Computer Vision*, pages 498–517. Springer, 2022.
- [17] Chenxia Li, Weiwei Liu, Ruoyu Guo, Xiaoting Yin, Kaitao Jiang, Yongkun Du, Yuning Du, Lingfeng Zhu, Baohua Lai, Xiaoguang Hu, et al. Pp-ocrv3: More attempts for the improvement of ultra lightweight ocr system. *arXiv preprint arXiv:2206.03001*, 2022.
- [18] Hang Li. Cdla: A chinese document layout analysis (cdla) dataset, 2021.
- [19] Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26763–26773, 2024.
- [20] Shuai Liu, Youmeng Li, and Jizeng Wei. Xy-cut++: Advanced layout ordering via hierarchical mask mechanism on a novel benchmark, 2025.
- [21] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12), December 2024.
- [22] Yuliang Liu, Biao Yang, Qiang Liu, Zhang Li, Zhiyin Ma, Shuo Zhang, and Xiang Bai. Textmonkey: An ocr-free large multimodal model for understanding document. *arXiv preprint arXiv:2403.04473*, 2024.
- [23] Yuliang Liu, Jiaxin Zhang, Dezheng Peng, Mingxin Huang, Xinyu Wang, Jingqun Tang, Can Huang, Dahua Lin, Chunhua Shen, Xiang Bai, et al. Spts v2: single-point scene text spotting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):15665–15679, 2023.
- [24] Nikolaos Livathinos, Christoph Auer, Maksym Lysak, Ahmed Nassar, Michele Dolfi, Panos Vagenas, Cesar Berrospi Ramis, Matteo Omenetti, Kasper Dinkla, Yusik Kim, et al. Docling: An efficient open-source toolkit for ai-driven document conversion. *arXiv preprint arXiv:2501.17887*, 2025.
- [25] Mahshad Mahdavi, Richard Zanibbi, Harold Mouchere, Christian Viard-Gaudin, and Utpal Garain. Icdar 2019 crohme+ tfd: Competition on recognition of handwritten mathematical expressions and typeset formula detection. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1533–1538. IEEE, 2019.
- [26] Mathpix. Mathpix snip: Convert images and pdfs to latex, docx, and more. <https://mathpix.com/>, 2025.
- [27] Mistral OCR. Mistral ocr: Free online ai ocr tool to extract text. <https://www.mistralocr.com/>, 2025.
- [28] Harold Mouchere, Christian Viard-Gaudin, Richard Zanibbi, and Utpal Garain. Icfhr 2014 competition on recognition of on-line handwritten mathematical expressions (crohme 2014). In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 791–796. IEEE, 2014.

- [29] Harold Mouchère, Christian Viard-Gaudin, Richard Zanibbi, and Utpal Garain. Icfhr2016 crohme: Competition on recognition of online handwritten mathematical expressions. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 607–612. IEEE, 2016.
- [30] Ahmed Nassar, Andres Marafioti, Matteo Omenetti, Maksym Lysak, Nikolaos Livathinos, Christoph Auer, Lucas Morin, Rafael Teixeira de Lima, Yusik Kim, A Said Gurbuz, et al. Smoldocling: An ultra-compact vision-language model for end-to-end multi-modal document conversion. *arXiv preprint arXiv:2503.11576*, 2025.
- [31] OleehyO. Texteller: An end-to-end formula recognition model. <https://github.com/OleehyO/TexTeller>, 2025.
- [32] OpenAI. Hello gpt-4o. <https://openai.com/index/hello-gpt-4o>, 2024.
- [33] Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, et al. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. *arXiv preprint arXiv:2412.07626*, 2024.
- [34] Inkit Padhi, Yair Schiff, Igor Melnyk, Mattia Rigotti, Youssef Mroueh, Pierre Dognin, Jerret Ross, Ravi Nair, and Erik Altman. Tabular transformers for modeling multivariate time series. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3565–3569. IEEE, 2021.
- [35] Vik Paruchuri. Marker, 2024.
- [36] Dezhi Peng, Xinyu Wang, Yuliang Liu, Jiaxin Zhang, Mingxin Huang, Songxuan Lai, Jing Li, Shenggao Zhu, Dahua Lin, Chunhua Shen, et al. Spts: single-point text spotting. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4272–4281, 2022.
- [37] Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S Nassar, and Peter Staar. Doclaynet: A large human-annotated dataset for document-layout segmentation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 3743–3751, 2022.
- [38] Jake Poznanski, Jon Borchardt, Jason Dunkelberger, Regan Huff, Daniel Lin, Aman Rangapur, Christopher Wilhelm, Kyle Lo, and Luca Soldaini. olmocr: Unlocking trillions of tokens in pdfs with vision language models. *arXiv preprint arXiv:2502.18443*, 2025.
- [39] Unstructured-IO. Unstructured: Open-source etl for complex document transformation. <https://github.com/Unstructured-IO/unstructured>, 2025.
- [40] Jianqiang Wan, Sibo Song, Wenwen Yu, Yuliang Liu, Wenqing Cheng, Fei Huang, Xiang Bai, Cong Yao, and Zhibo Yang. Omniparser: A unified framework for text spotting key information extraction and table recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15641–15653, 2024.
- [41] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, et al. Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems*, 37:107984–108011, 2024.
- [42] Bin Wang, Zhuangcheng Gu, Guang Liang, Chao Xu, Bo Zhang, Botian Shi, and Conghui He. Unimernet: A universal network for real-world mathematical expression recognition. *arXiv preprint arXiv:2404.15254*, 2024.
- [43] Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, et al. Mineru: An open-source solution for precise document content extraction. *arXiv preprint arXiv:2409.18839*, 2024.
- [44] Zilong Wang, Yiheng Xu, Lei Cui, Jingbo Shang, and Furu Wei. Layoutreader: Pre-training of text and layout for reading order detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4735–4744, 2021.
- [45] Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, et al. General ocr theory: Towards ocr-2.0 via a unified end-to-end model. 2024.

- [46] Renqiu Xia, Song Mao, Xiangchao Yan, Hongbin Zhou, Bo Zhang, Haoyang Peng, Jiahao Pi, Daocheng Fu, Wenjie Wu, Hancheng Ye, et al. Docgenome: An open large-scale scientific document benchmark for training and testing multi-modal large language models. *arXiv preprint arXiv:2406.11633*, 2024.
- [47] Ye Yuan, Xiao Liu, Wondimu Dikubab, Hui Liu, Zhilong Ji, Zhongqin Wu, and Xiang Bai. Syntax-aware network for handwritten mathematical expression recognition. *arXiv preprint arXiv:2203.01601*, 2022.
- [48] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16965–16974, 2024.
- [49] Zhiyuan Zhao, Hengrui Kang, Bin Wang, and Conghui He. Doclayout-yolo: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception. *arXiv preprint arXiv:2410.12628*, 2024.
- [50] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Livia Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sclang: Efficient execution of structured language model programs. *Advances in Neural Information Processing Systems*, 37:62557–62583, 2024.
- [51] Xinyi Zheng, Douglas Burdick, Lucian Popa, Xu Zhong, and Nancy Xin Ru Wang. Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 697–706, 2021.
- [52] Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. Image-based table recognition: data, model, and evaluation. *arXiv preprint arXiv:1911.10683*, 2019.
- [53] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022. IEEE, Sep. 2019.