SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

November 2026, Vol. 69, Iss. 11, 212105:1–212105:12
https://doi.org/10.1007/s11432-026-5121-6

MonkeyOCR: document parsing with a structure-recognition-relation triplet paradigm

Zhang LI[1], Yuliang LIU[1*], Qiang LIU[2], Zhiyin MA[1], Ziyang ZHANG[1], Shuo ZHANG[1], Biao YANG[1], Zidun GUO[1], Jiarui ZHANG[2], Xinyu WANG[1] & Xiang BAI[3*]

[1]*School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China*
[2]*Kingsoft Office, Wuhan 430075, China*
[3]*School of Software Engineering, Huazhong University of Science and Technology, Wuhan 430074, China*

Received 25 February 2026/Revised 15 May 2026/Accepted 14 July 2026/Published online 3 November 2026

Abstract We introduce MonkeyOCR, a document parsing model 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 and avoids the inefficiencies of processing full pages with giant end-to-end models. 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 structure detection, content recognition, and relation prediction. To support this paradigm, we present MonkeyDoc, a comprehensive dataset with 4.5 million bilingual instances spanning over ten document types, which addresses the limitations of existing datasets that often focus on a single task, language, or document type. Leveraging the SRR paradigm and MonkeyDoc, we trained a 3B-parameter document foundation model. We further identify parameter redundancy in this model and propose contiguous parameter degradation (CPD), enabling the construction of models from 0.6B to 1.2B parameters that run faster with an acceptable performance drop. MonkeyOCR achieves state-of-the-art performance, surpassing previous open-source and closed-source methods, including Gemini 2.5 Pro. Additionally, the model can be efficiently deployed for inference on a single RTX 3090 GPU. Code and models will be released at https://github.com/Yuliang-Liu/MonkeyOCR.

Keywords document parsing, large multimodal model, structure detection, content recognition, relation prediction

Citation Li Z, Liu Y L, Liu Q, et al. MonkeyOCR: document parsing with a structure-recognition-relation triplet paradigm. Sci China Inf Sci, 2026, 69(11): 212105, https://doi.org/10.1007/s11432-026-5121-6

1 Introduction

Document parsing aims to systematically convert the complex multimodal content (including texts, tables, images, and formulas) within various documents (such as scanned images and PDF files) into structured information. This technology has a wide range of applications, including smart office solutions, intelligent education systems, digitization of medical records and judicial documents, and the preservation of historical documents.

Recent mainstream document parsing methods can be broadly categorized into pipeline-based toolchains and end-to-end models. Pipeline-based toolchains, such as MinerU [1] and PPStruct-V3 [2], break down the complex parsing task into fine-grained sub-tasks, with each sub-task handled by specialized models. This modular approach allows each component to be optimized independently. In contrast, giant end-to-end models [3] directly input entire document page images into a unified model for holistic content recognition, resulting in a streamlined workflow.

Although existing document parsing approaches have achieved remarkable progress, they still suffer from inherent limitations. First, pipeline-based methods suffer from cumulative error caused by the combination of multiple fine-grained tools. These methods typically follow a sequential process: layout detection, text and formula detection, instance-level recognition, element merging, and finally reading order reconstruction. As shown in Figure 1, imprecise formula detection results in partial cropping of characters from the previous line, leading to erroneous recognition outcomes, such as the inclusion of extraneous superscripts. Second, end-to-end models often face limitations in both performance and efficiency when dealing with lengthy contexts. Document pages are typically of high resolution and contain densely packed information, leading to extremely long input and output sequences. Prior studies [4, 5] have shown that large multimodal models are more prone to hallucinations when processing long contexts. Moreover, the quadratic computational cost of attention makes the processing of long sequences increasingly expensive.

* Corresponding author (email: ylliu@hust.edu.cn, xbai@hust.edu.cn)

© The Author(s) 2026. This article is published with open access at link.springer.com.

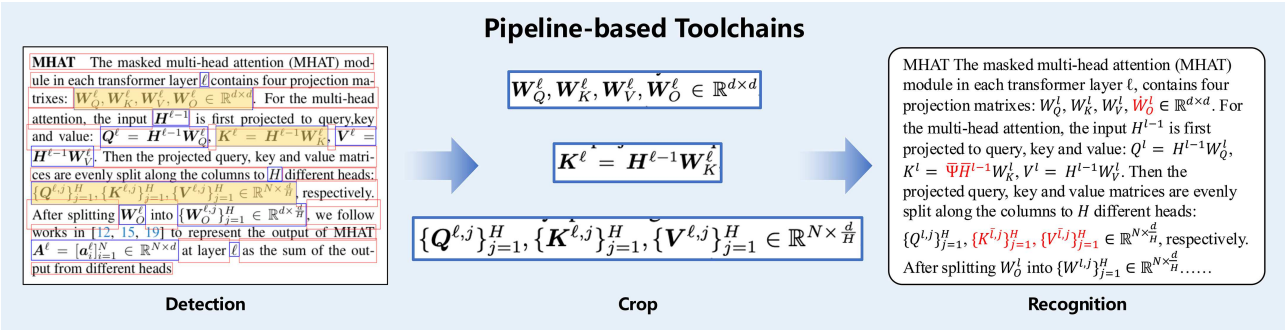


Figure 1 (Color online) A typical example of cumulative error in pipeline-based toolchains.

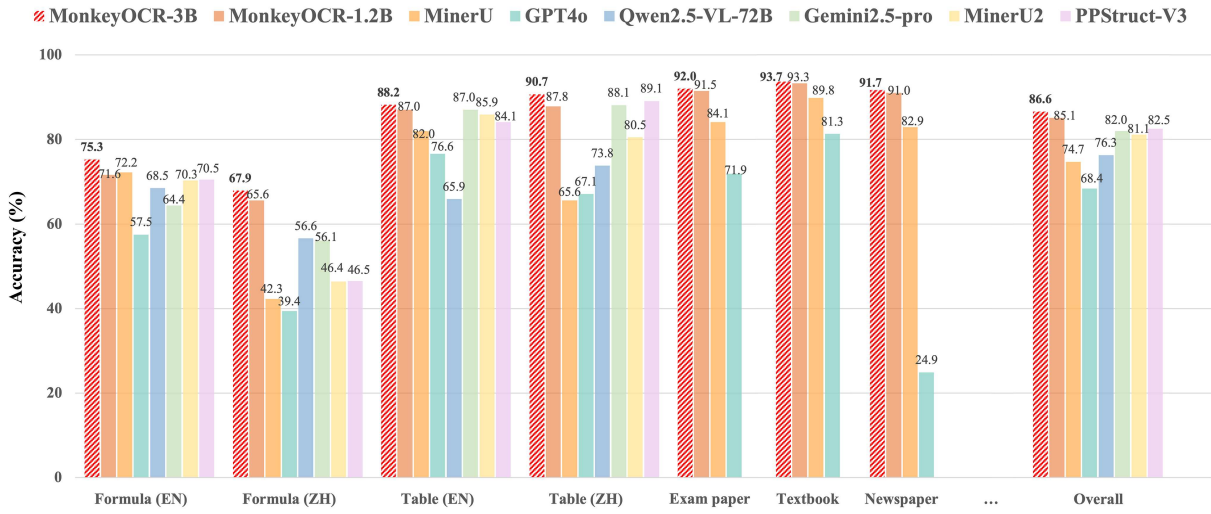


Figure 2 (Color online) Performance comparison of MonkeyOCR and other SOTA models on OmniDocBench [6]. Accuracy is calculated as 1 minus the normalized edit distance (NED).

To achieve a balance between accuracy and speed, we propose MonkeyOCR, which adopts a "structure-recognition-relation" (SRR) triplet paradigm for document parsing, as shown in Figure 2. Specifically, we first perform structure detection to identify different regions within the document, such as text blocks, tables, formulas, and images. Then, to handle multiple recognition tasks with a unified model, we trained a large multimodal model to perform block-level parallel end-to-end recognition of these detected regions. Finally, the coordinates and categories of these regions are fed into a block-level relation prediction model to determine the reading order. Based on the reading order, the results of content recognition are then combined to produce the final document content. Compared with pipeline-based methods that detect fine-grained text instances, MonkeyOCR performs detection at the element-block level. This design is more robust and efficient, since the spacing between element blocks is typically much larger than the line spacing within text regions (5.2 mm vs. 1.5 mm). In our statistics, MinerU produces an average of 97.2 detection boxes per page, whereas MonkeyOCR generates only 19.5 boxes per page, thereby mitigating error accumulation across stages. Moreover, our framework requires only three models, while conventional pipeline-based methods usually rely on at least seven models, together with complex post-processing for merging text instances and reconstructing document structure. This streamlined design effectively reduces error accumulation and improves parsing efficiency.

Existing document parsing datasets often focus on a single task (e.g., table recognition or layout analysis), a specific document type (e.g., academic papers), or a single language, which limits the development of robust document parsing models. To address these limitations, we developed a comprehensive data generation pipeline that integrates meticulous manual annotation, programmatic data synthesis, and advanced model-driven automatic labeling, resulting in MonkeyDoc dataset. The dataset contains 4.5 million bilingual instances, covering five document parsing tasks, involving more than ten document types. Building on the SRR paradigm and the MonkeyDoc dataset, we trained a 3B-parameter document foundation model. We further observed parameter redundancy in this model and, based on experimental findings, empirically propose contiguous parameter degradation (CPD), which continuously skips sequences of intermediate layers to construct lightweight models (0.6B–1.2B parameters) that

achieve comparable performance with faster inference. Moreover, our results show that parameter reduction has minimal impact on simple tasks, such as text recognition, but substantially affects more complex tasks, such as table recognition. Compared to pipeline-based methods, MonkeyOCR reduces the cumulative errors caused by multi-tool combination, outperforming the best pipeline method PPStruct-V3 by 4.1%. It also surpasses the giant end-to-end model Qwen2.5-VL-72B and the best closed-source model Gemini 2.5 Pro, achieving 2.09× faster inference than the end-to-end baseline. Furthermore, MonkeyOCR-1.2B offers a 34% speed improvement over the 3B model with only a 1.5% performance drop. We summarize the contributions as follows.

• We propose the SRR paradigm, decomposing document parsing into structure detection, content recognition, and relation prediction. This design reduces cumulative errors caused by multi-tool integration and avoids the inefficiencies of processing full pages with end-to-end models.

• We present MonkeyDoc, a comprehensive dataset with 4.5 million bilingual instances covering five document parsing tasks across more than ten document types, addressing the limitations of existing datasets that often focus on a single task, language, or document type.

• We propose a 3B document foundation model based on the SRR paradigm and MonkeyDoc, and introduce CPD to prune redundant layers. CPD enables the construction of lightweight models (0.6B–1.2B parameters) that achieve comparable performance while offering faster inference.

2 Related work

2.1 Pipeline-based toolchains

Pipeline-based approaches [1, 7] decompose the task of document parsing into multiple sequential steps, such as layout analysis [8–10], relation prediction [11], optical character recognition (OCR) [12–14], formula recognition [15, 16], formula detection, and table recognition [17]. Each step employs a specialized model to handle a specific sub-task. For instance, CRNN [13] combines convolutional feature extraction with recurrent sequence modeling for text recognition. Docling [7] implements a linear processing pipeline to parse PDF files, extract page content (including OCR, layout, and table information), and generate structured documents in JSON or Markdown format. Marker[1)] relies on the Surya[2)] OCR toolkit for OCR, layout, reading order, and table recognition, with optional LLM modules for cross-page table merging and inline math parsing. Similarly, MinerU and PPStruct-V3 adopt a pipeline of multiple models for layout detection [18], formula detection[3)] and recognition [16], OCR [12], table recognition [17, 19], and relation prediction [11, 20], and integrate these outputs to complete document parsing.

2.2 End-to-end models

End-to-end models streamline document parsing by directly processing full-page images into structured outputs, eliminating the need for multiple task-specific models. Donut [21] introduces the first OCR-free framework for document understanding, while Nougat [22] adopts a similar architecture to convert document images into Markdown. ABCNet [23] employs adaptive Bezier curves to jointly detect and recognize arbitrarily shaped scene text. The SPTS series [24, 25] presents a unified framework for text detection and recognition, and OmniParser [26] extends it to text spotting, key information extraction, and table recognition, enabling robust parsing across both natural scenes and documents. Recently, large multimodal models [3, 27–32] trained on massive datasets have achieved remarkable progress in end-to-end document understanding. Building on Qwen2.5-VL [3], models such as OLMOCR [33], OCRFlux[4)], and Nanonets-OCR-s[5)] are further trained on large corpora of PDF pages, enhancing the document parsing capabilities of general-purpose multimodal models. DeepSeek-OCR-3B [34] explores visual token compression techniques for efficient document parsing. UniRec [35] introduces a lightweight unified recognition model with only 0.1B parameters, supporting multi-level text and formula recognition and achieving a 2–9× inference speedup. However, it is limited to text and formula recognition and does not support table parsing. Different from UniRec, which is a specially designed lightweight recognition model, our CPD serves as a compression framework for existing recognition models, and is therefore complementary to lightweight model design.

1) https://github.com/datalab-to/marker.
2) https://github.com/datalab-to/surya.
3) https://github.com/ultralytics/ultralytics.
4) https://github.com/chatdoc-com/OCRFlux.
5) https://huggingface.co/nanonets/Nanonets-OCR-s.

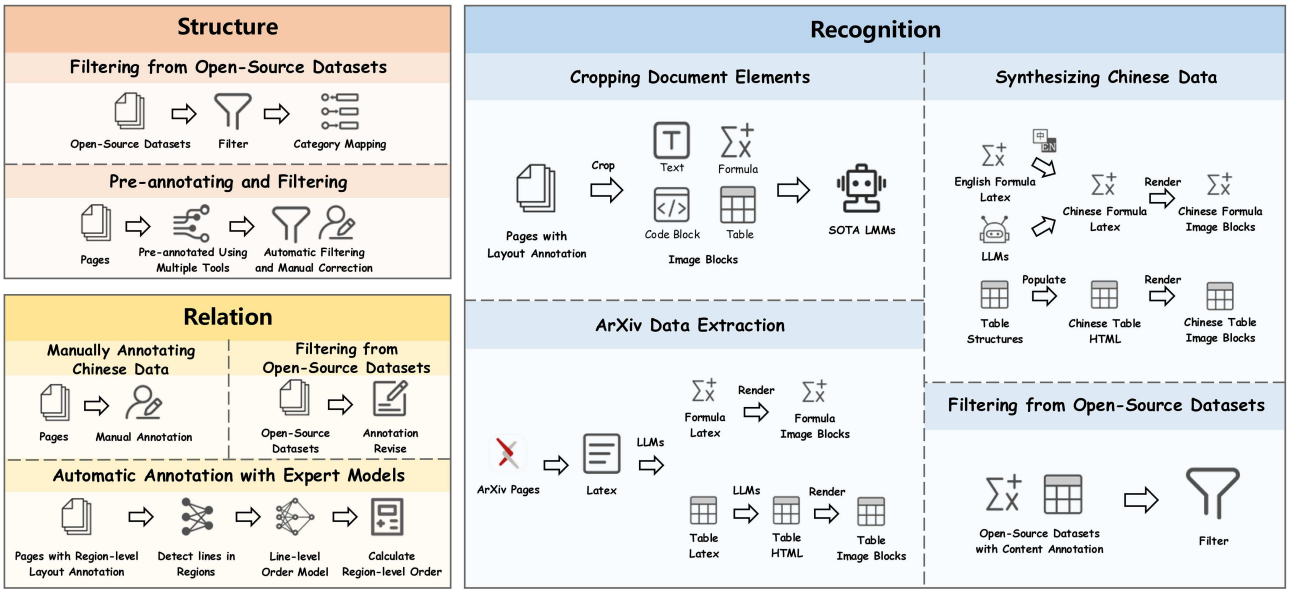


Figure 3 (Color online) The data generation pipeline of MonkeyDoc.

Table 1 Comparison of MonkeyDoc with other document parsing datasets.

Dataset	Document type	Supporting tasks					Language	
		Layout detection	Relation prediction	Formula recognition	Table recognition	Text recognition	EN	ZH
M^{6}Doc [36], DocLayNet [38]	6, 2	✓					✓	✓
CDLA[a)]	1	✓						✓
D4LA [37]	5	✓					✓	
UniMER-1M [16], TexTeller[b)]	–			✓				
FinTabNet [39], PubTabNet [40]	–				✓			
DocGenome [19]	1	✓	✓	✓	✓	✓	✓	
MonkeyDoc	**>10**	✓	✓	✓	✓	✓	✓	✓

a) https://github.com/buptlihang/CDLA.
b) https://github.com/OleehyO/TexTeller.

2.3 Document parsing dataset

Document parsing encompasses fine-grained tasks such as layout detection, relation prediction, and content recognition. A variety of datasets have been developed to support these tasks, each focusing on specific document types, languages or sub-tasks. For layout detection, datasets such as M^{6}Doc [36], CDLA, D4LA [37], and DocLayNet [38] provide annotations of the positions of structural elements (e.g., text blocks, tables) across diverse document types. For content recognition, table recognition datasets such as FinTabNet [39] and PubTabNet [40] provide large-scale structural and content annotations, while UniMER-1M [16] and HME-100k [41] focus on mathematical expression recognition in printed and handwritten forms. In addition, OCRBench [42] provides a comprehensive benchmark for evaluating the text recognition and understanding capabilities of large multimodal models. For relation prediction, ReadingBank [11] provides word-level reading order annotations for 500k documents. Additionally, DocGenome [19] provides annotations for 500k English papers from ArXiv, covering major document parsing tasks.

3 MonkeyDoc dataset

To train MonkeyOCR, we developed a complete data generation pipeline and built a comprehensive document parsing dataset, MonkeyDoc. As shown in Table 1, the dataset covers more than ten document types, supports both Chinese and English, and spans the entire document parsing process. The overall data generation pipeline is illustrated in Figure 3. More details can be found in the Supporting information.

For structure detection, we collected raw annotations from existing open-source datasets [19, 36–38], removed nested bounding boxes, and selected samples with richer structural elements to enhance data diversity. Moreover,

considering the scarcity of Chinese samples in open-source datasets, we additionally collected over 300k pages of Chinese documents, including financial reports, textbooks, academic papers, etc. These pages were pre-annotated using multiple tools [2,18], followed by rule-based automatic filtering and manual correction and validation to ensure annotation accuracy and consistency.

For content recognition, we cropped a total of 2.5 million sub-images of document elements based on the structure detection results, which were then annotated using Qwen2.5-VL-72B [3] and Gemini 2.5 Pro [43]. To further expand our datasets for table and formula recognition, we employed a rule-based method to filter out erroneous samples from the open-source datasets PubTabNet [40] and UniMER-1M [16]. Addressing the scarcity of Chinese data, we populated existing table structures with Chinese characters and rendered them into new table images. For formulas, we designed prompts to guide an LLM in generating commonly used Chinese formulas across various domains and also used the LLM to translate existing English formulas into Chinese, which were subsequently rendered. To enhance the diversity and authenticity of our table and formula data, we scraped LaTeX source from ArXiv, used an LLM to filter out irrelevant content, converted the tables to HTML format, and then rendered them into images.

For relation prediction, our goal is to construct a block-level reading order dataset that bridges the gap between existing block-level structure detection and line-level relation prediction methods. First, we refined the existing open-source dataset [19] by linking tables and images to their corresponding captions and removing low-quality samples to improve annotation accuracy. Second, to address the scarcity of Chinese documents, we manually annotated a variety of Chinese document types, enriching the dataset's diversity. Finally, to fully utilize the large amount of structure detection data with layout annotations but without reading order information, we adopted an automatic labeling strategy. This strategy first employs an OCR tool [12] to detect text lines within each block, then uses a line-level reading order model [11] to predict their sequence. The block-level reading order is then inferred by calculating the average reading order of all text lines within the block.

To evaluate the quality of the labeled data, we randomly sampled 100 instances from each of the text, table, and formula categories for manual validation. The generated annotations achieve 1-NED scores of 99.6, 99.96, and 99.1, respectively, indicating high annotation quality. Overall, approximately 3% of the dataset has been manually verified to ensure annotation reliability.

4 MonkeyOCR

Figure 4 illustrates the overall pipeline of MonkeyOCR. First, the input image is fed into the structure detection model to obtain bounding boxes and categories for each document element. Based on these bounding boxes, the corresponding text, table, and formula regions are cropped and sent to a large multimodal model (LMM) for content recognition. Meanwhile, the coordinates and categories of the detected elements are passed to the relation prediction model to determine the reading order of the document elements. Finally, the recognized results are integrated according to the predicted reading order to produce the final structured document. The proposed SRR paradigm simplifies traditional pipeline-based methods into three unified models, thereby mitigating the cumulative errors caused by multiple tool combinations. Moreover, by processing document blocks in parallel, our method avoids the long context lengths associated with full-document processing, leading to substantial gains in speed. We also found parameter redundancy in large multimodal models and proposed the contiguous parameter degradation method, which enables faster inference with minimal performance degradation.

4.1 Structure detection

We employ a DETR-based [8] model for structure detection, which consists of a backbone, an encoder, a query selection module, as well as a decoder and prediction heads. Given a document image $I \in \mathbb{R}^{H\times W\times 3}$, the backbone and encoder extract visual features

$$F_0 = \text{Backbone}(I), F_{\text{enc}} = \text{Encoder}(F_0), \tag{1}$$

where $F_{\text{enc}} \in \mathbb{R}^{N\times C}$ represents the encoded feature tokens.

To estimate the likelihood of each token corresponding to a foreground region, a lightweight classification head is applied to F_{enc}

$$s = \text{SiLU}(\text{BN}(\text{Conv}_{1\times 1}(F_{\text{enc}}))), \tag{2}$$

where s represents the foreground probability of each token, indicating its likelihood of belonging to meaningful elements (e.g., text, table, figure).

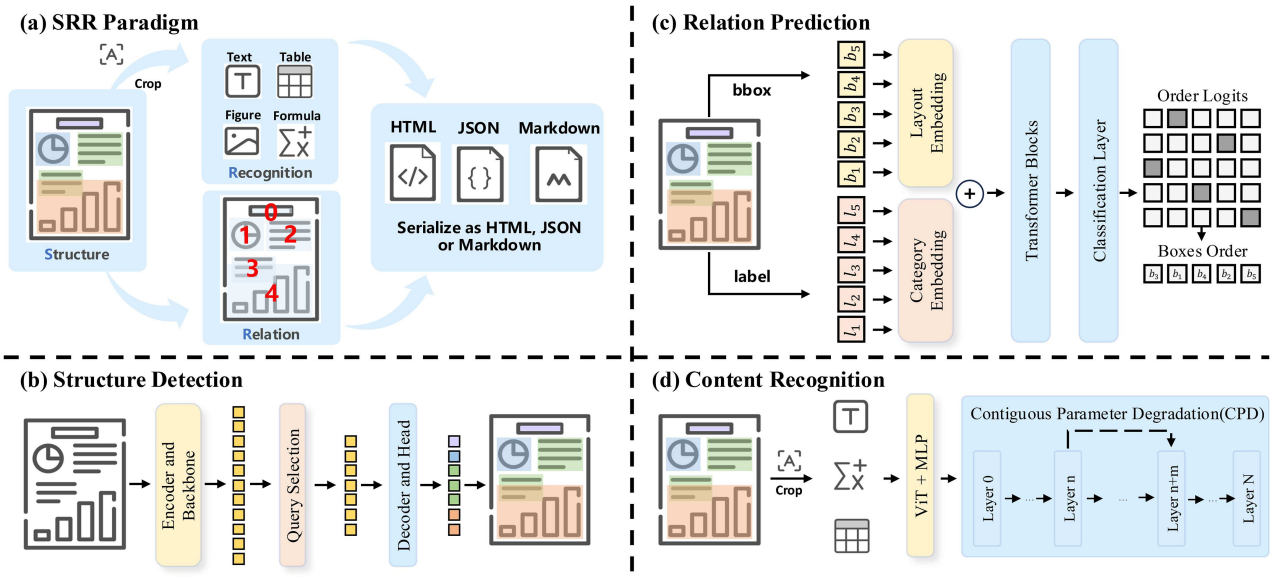


Figure 4 (Color online) The overall pipeline of MonkeyOCR.

Based on these scores, the query selection module picks the top-K most informative features

$$Q_0 = \text{TopK}(F_{\text{enc}}, s, K), \tag{3}$$

effectively filtering background noise and providing high-quality queries for the decoder.

Finally, the decoder refines the selected queries through iterative attention, followed by task-specific heads to produce the final layout predictions.

$$y = \text{Head}(\text{Decoder}(Q_0, F_0)), \tag{4}$$

where each prediction $y_i = (b_i, l_i)$ includes a bounding box b_i and a category label l_i.

4.2 Content recognition

Given the detected bounding boxes $B = \{b_1, b_2, \ldots, b_n\}$, each region is first cropped from the original document image I to obtain sub-images

$$I^i_{\text{crop}} = \text{Crop}(I, b_i). \tag{5}$$

Each cropped image is preserved at its original resolution and then encoded into a sequence of visual tokens through a vision encoder followed by an MLP projection

$$T^i_v = \text{MLP}(\text{VisionEnc}(I^i_{\text{crop}})). \tag{6}$$

For each block, a type-specific prompt p_{l_i} is selected according to its predicted category l_i (e.g., text, table, or formula). Finally, the visual tokens and the corresponding prompt tokens are jointly fed into an LLM to produce the final content predictions

$$C = \text{LLM}(\{T^1_v, \ldots, T^n_v\}, \{p_{l_1}, \ldots, p_{l_n}\}), \tag{7}$$

where $C = \{c_1, c_2, \ldots, c_n\}$ denotes the textual representations of the detected document elements.

4.3 Relation prediction

We build a block-level category-aware relation prediction model, which consists of embedding layers, transformer blocks and a classification layer. Given a bounding box $b_i = (x_1, y_1, x_2, y_2)$ and its width and height, its positional embedding is computed as

$$P_i = \text{Concat}\big(E(p) \mid p \in \{x_1, y_1, x_2, y_2, w, h\}\big), \tag{8}$$

where $E(\cdot)$ denotes separate embedding layers. Different types of document elements typically exhibit distinct semantic priorities for reading order. To explicitly capture this, we propose a category-aware embedding and combine it with positional encoding to construct more discriminative input features

$$F_i = P_i + E_{\text{c}}(l_i), \tag{9}$$

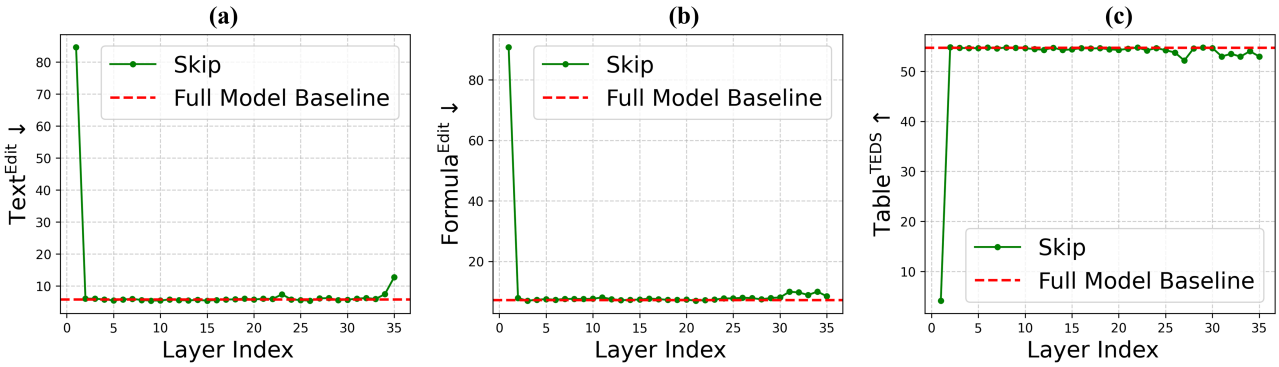


Figure 5 (Color online) Performance of MonkeyOCR-3B under layer skipping. (a) Text recognition; (b) formula recognition; (c) table recognition. Green denotes skipping layer N, and red denotes no skipping.

where l_i denotes the category of the element.

The sequence of features $[F_1, \ldots, F_N]$ is processed by transformer layers to model contextual relationships, followed by a classification layer producing the reading order logits

$$L = \text{Classifier}(\text{Transformer}([F_1, \ldots, F_N])), \tag{10}$$

where L represents the predicted logits for each element's position. The final reading order is obtained via iterative greedy decoding: each element is first assigned to its highest-scoring position; conflicts are resolved by retaining the element with the highest logit and reassigning others to their next-best positions. This process repeats until a valid permutation is obtained.

4.4 Contiguous parameter degradation

In the field of document intelligence, document parsing serves as an upstream task where processing speed is critical. Since models with fewer parameters generally achieve faster inference, we pose the following question: are all layers in a large multimodal model truly necessary? To investigate this, we randomly selected 50 samples from each of the three tasks: text recognition, formula recognition, and table recognition, for experimentation. We systematically pruned the LLM one layer at a time and evaluated the performance on each task. The results, shown in Figure 5, reveal that removing the first layer of the LLM leads to a notable performance drop, whereas pruning other layers results in relatively minor degradation. This observation suggests that not all layers in the LLM are indispensable, and that certain parameters may be redundant. However, how to effectively prune the model remains a question. To this end, we conduct a systematic ablation study on different pruning strategies in Subsection 5.3. Based on the results, we empirically propose contiguous parameter degradation, which directly prunes m contiguous middle layers of the LLM followed by fine-tuning. We hypothesize that adjacent layers in the LLM carry continuous and tightly coupled information. By performing contiguous pruning primarily on the middle layers, we can maximally preserve the continuity and semantic relationships between neighboring layers, thereby maintaining performance.

4.5 Implementation details

MonkeyOCR is initialized from pretrained models [3, 11, 44] and subsequently undergoes large-scale training on MonkeyDoc. 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 on 32 A800 GPUs.

5 Experiments

To validate the effectiveness of MonkeyOCR, we conducted a comprehensive comparison with both open-source and closed-source methods on OmniDocBench [6]. All document images are processed at their original resolutions in our method, without artificial upsampling. For all compared methods, we strictly follow the official OmniDocBench evaluation protocol, including the prompt settings for general vision-language models (VLMs).

Table 2 The end-to-end evaluation results of different tasks on OmniDocBench. The best results are highlighted in bold.

Model type	Method	Overall$^{\text{Edit}}\downarrow$		Text$^{\text{Edit}}\downarrow$		Formula$^{\text{Edit}}\downarrow$		Table$^{\text{TEDS}}\uparrow$		Table$^{\text{Edit}}\downarrow$		Read order$^{\text{Edit}}\downarrow$	
		EN	ZH	EN	ZH	EN	ZH	EN	ZH	EN	ZH	EN	ZH
Pipeline tools	MinerU [1]	0.150	0.357	0.061	0.215	0.278	0.577	78.6	62.1	0.180	0.344	0.079	0.292
	Marker	0.336	0.556	0.080	0.315	0.530	0.883	67.6	49.2	0.619	0.685	0.114	0.340
	Mathpix[a)]	0.191	0.365	0.105	0.384	0.306	0.454	77.0	67.1	0.243	0.320	0.108	0.304
	Docling [7]	0.589	0.909	0.416	0.987	0.999	1	61.3	25.0	0.627	0.810	0.313	0.837
	Pix2Text[b)]	0.320	0.528	0.138	0.356	0.276	0.611	73.6	66.2	0.584	0.645	0.281	0.499
	Unstructured[c)]	0.586	0.716	0.198	0.481	0.999	1	0	0.06	1	0.998	0.145	0.387
	OpenParse[d)]	0.646	0.814	0.681	0.974	0.996	1	64.8	27.5	0.284	0.639	0.595	0.641
	PPStruct-V3 [2]	0.145	0.206	0.058	0.088	0.295	0.535	77.2	83.9	0.159	0.109	0.069	**0.091**
Expert VLMs	GOT-OCR [27]	0.287	0.411	0.189	0.315	0.360	0.528	53.2	47.2	0.459	0.520	0.141	0.280
	Nougat [22]	0.452	0.973	0.365	0.998	0.488	0.941	39.9	0	0.572	1.000	0.382	0.954
	Mistral OCR[e)]	0.268	0.439	0.072	0.325	0.318	0.495	75.8	63.6	0.600	0.650	0.083	0.284
	OLMOCR [33]	0.326	0.469	0.097	0.293	0.455	0.655	68.1	61.3	0.608	0.652	0.145	0.277
	SmolDocling [46]	0.493	0.816	0.262	0.838	0.753	0.997	44.9	16.5	0.729	0.907	0.227	0.522
	Dolphin [47]	0.206	0.306	0.107	0.197	0.447	0.580	77.3	67.2	0.180	0.285	0.091	0.162
	MinerU2.0 [1]	0.139	0.240	0.047	0.109	0.297	0.536	82.5	79.0	0.141	0.195	0.069	0.118
	OCRFlux-3B	0.195	0.281	0.064	0.183	0.379	0.613	71.6	81.3	0.253	0.139	0.086	0.187
	Nanonets-OCR-3B	0.283	0.295	0.134	0.231	0.518	0.546	76.8	79.4	0.343	0.201	0.135	0.200
	DeepSeek-OCR-3B-Large [34]	0.138	0.208	0.054	0.143	0.277	0.461	–	–	0.152	0.104	0.067	0.123
General VLMs	GPT4o [45]	0.233	0.399	0.144	0.409	0.425	0.606	72.0	62.9	0.234	0.329	0.128	0.251
	Qwen2.5-VL-7B [3]	0.312	0.406	0.157	0.228	0.351	0.574	76.4	72.2	0.588	0.619	0.149	0.203
	InternVL3-8B [32]	0.314	0.383	0.134	0.218	0.417	0.563	66.1	73.1	0.586	0.564	0.118	0.186
	Qwen2.5-VL-72B [3]	0.214	0.261	0.092	0.180	0.315	0.434	82.9	83.9	0.341	0.262	0.106	0.168
	Gemini 2.5 Pro [43]	0.148	0.212	0.055	0.168	0.356	0.439	**85.8**	86.4	0.130	0.119	**0.049**	0.121
	MonkeyOCR-1.2B	0.133	0.166	0.047	0.087	0.284	0.344	83.0	85.8	0.130	0.122	0.069	0.112
	MonkeyOCR-3B	**0.118**	**0.151**	**0.043**	**0.077**	**0.247**	**0.321**	84.2	**89.5**	**0.118**	**0.093**	0.066	0.113

a) https://mathpix.com.
b) https://github.com/breezedeus/pix2text.
c) https://github.com/Unstructured-IO/unstructured.
d) https://github.com/Filimoa/open-parse.
e) https://www.mistralocr.com.

5.1 Comparison on different tasks

Document parsing encompasses a variety of sub-tasks, including text recognition, formula recognition, table recognition, relation prediction, and more. To evaluate MonkeyOCR's performance across these tasks, we compared it with widely-used methods on OmniDocBench [6], including pipeline tools [1, 7], expert VLMs [27], closed-source general VLMs [45], and open-source general VLMs [3, 32]. As shown in Table 2, MonkeyOCR achieves the best overall performance on both Chinese and English document parsing tasks, attaining state-of-the-art results on 7 out of 10 sub-tasks. Specifically, our 3B model surpasses the best pipeline-based method PPStruct-V3, achieving a 13.1% improvement in formula recognition and outperforming the giant closed-source end-to-end model Gemini 2.5 Pro. Moreover, our 1.2B model attains the second-best overall performance, outperforming both DeepSeek-OCR-3B and Nanonets-OCR.

5.2 Comparison across document types

As shown in Table 3, MonkeyOCR achieves the best overall performance. In particular, the MonkeyOCR-3B model outperforms both large-scale general VLMs and pipeline-based systems, exceeding InternVL3-8B [32] by 12.1% and MinerU [1] by 13.9% in overall performance. Remarkably, MonkeyOCR-1.2B model also delivers highly competitive results, ranking second overall with a score of 0.078, only 1.1% behind the 3B model. Beyond overall performance, MonkeyOCR demonstrates strong robustness across different document categories, with the 3B model achieving the best results on six types of documents. These results indicate that our model performs well across different document types.

Table 3 The end-to-end text recognition performance on OmniDocBench across 9 PDF page types.

Model type	Model	Book	Slides	Financial report	Textbook	Exam paper	Magazine	Academic papers	Notes	Newspaper	Overall
Pipeline tools	MinerU [1]	0.055	0.124	0.033	0.102	0.159	0.072	**0.025**	0.984	0.171	0.206
	Marker	0.074	0.340	0.089	0.319	0.452	0.153	0.059	0.651	0.192	0.274
	Mathpix	0.131	0.220	0.202	0.216	0.278	0.147	0.091	0.634	0.690	0.300
Expert VLMs	GOT-OCR [27]	0.111	0.222	0.067	0.132	0.204	0.198	0.179	0.388	0.771	0.267
	Nougat [22]	0.734	0.958	1.000	0.820	0.930	0.830	0.214	0.991	0.871	0.806
	Dolphin [47]	0.091	0.131	0.057	0.146	0.231	0.121	0.074	0.363	0.307	0.177
	OCRFlux-3B	0.068	0.125	0.092	0.102	0.119	0.083	0.047	0.223	0.536	0.149
General VLMs	GPT4o [45]	0.157	0.163	0.348	0.187	0.281	0.173	0.146	0.607	0.751	0.316
	Qwen2.5-VL-7B [3]	0.148	**0.053**	0.111	0.137	0.189	0.117	0.134	0.204	0.706	0.205
	InternVL3-8B [32]	0.163	0.056	0.107	0.109	0.129	0.100	0.159	0.150	0.681	0.188
	MonkeyOCR-1.2B	**0.048**	0.079	**0.014**	0.067	0.085	**0.057**	0.042	0.184	0.090	0.078
	MonkeyOCR-3B	**0.048**	0.068	0.015	**0.063**	**0.080**	**0.057**	0.040	**0.120**	**0.083**	**0.067**

Table 4 Ablation study on the proposed SRR paradigm and the MonkeyDoc dataset.

Method	$\text{Text}^{\text{Edit}}\downarrow$	$\text{Formula}^{\text{Edit}}\downarrow$	$\text{Table}^{\text{TEDS}}\uparrow$	$\text{Order}^{\text{Edit}}\downarrow$	$\text{Overall}^{\text{Edit}}\downarrow$
Baseline	0.349	0.525	66.2	0.341	0.461
+SRR	0.071	0.333	81.0	0.157	0.175
+MonkeyDoc	**0.060**	**0.284**	**86.9**	**0.090**	**0.135**

Table 5 Impact of SRR on error accumulation. "TD" and "FD" denote the text and formula detection models.

Recognition model	Pipeline	$\text{Text}^{\text{Edit}}\downarrow$		$\text{Formula}^{\text{Edit}}\downarrow$		$\text{Overall}^{\text{Edit}}\downarrow$	
		EN	ZH	EN	ZH	EN	ZH
Qwen2.5-VL	SRR	0.051	0.077	0.296	0.371	0.147	0.168
	+TD&FD	0.070	0.112	0.297	0.598	0.151	0.235
Ours	SRR	0.043	0.077	0.247	0.321	0.118	0.151
	+TD&FD	0.067	0.137	0.270	0.594	0.131	0.235

5.3 Ablation study

Ablation study on SRR and MonkeyDoc. We conducted experiments to systematically evaluate the effectiveness of the proposed SRR paradigm and the impact of training on the MonkeyDoc dataset. The baseline is the original Qwen2.5-VL-3B model with end-to-end prediction. In SRR, we use PP-DocLayoutV2 [44] for structure detection. As shown in Table 4, SRR significantly enhances the zero-shot document parsing performance of large multimodal models, achieving an average overall improvement of 28.6%, including a 27.8% gain in text recognition. This improvement can be attributed to SRR's ability to mitigate interference from long contexts during inference. Moreover, after training on the MonkeyDoc dataset, the overall performance increases by 4%, with a 5.9% improvement in table recognition and a 6.7% improvement in reading order, demonstrating the effectiveness of the MonkeyDoc dataset.

Impact of SRR paradigm on error accumulation. To evaluate the effect of the proposed SRR paradigm on cumulative errors, we extended MonkeyOCR by incorporating text and formula detection models following the MinerU pipeline. Under this setup, text blocks are no longer fed directly into the LMM for recognition. Instead, word-level bounding boxes are first obtained via the detection models, and the content within each box is then recognized by the LMM, with the results subsequently merged. As shown in Table 5, introducing additional detection models results in a 2.7% decrease in text recognition performance for Qwen2.5-VL and a 4.2% decrease for our model. Similarly, formula recognition performance drops by 11.4% for Qwen2.5-VL and 14.8% for our model. These results demonstrate the effectiveness of SRR in mitigating cumulative errors.

Analysis of inference speed. We randomly sampled 300 pages of PDFs from ArXiv to analyze the inference speed of the models, with the results summarized in Table 6. Baseline-3B corresponds to Qwen2.5-VL-3B model finetuned on MonkeyDoc, which performs end-to-end prediction directly. The results show that MonkeyOCR-1.2B achieves a 34% speedup over MonkeyOCR-3B, while incurring only a 1.5% drop in performance. Furthermore, by enabling parallel block-wise prediction, MonkeyOCR-3B runs 2.09× faster than the fully end-to-end baseline,

Table 6 Inference speed (pages/s) comparison on different GPUs.

Model	3090	4090	A6000	H800	Avg.
MonkeyOCR-3B	0.497	1.006	0.609	0.897	0.752
MonkeyOCR-1.2B	0.677	1.436	0.825	1.101	1.010
Baseline-3B	0.244	0.387	0.285	0.518	0.359

Table 7 Ablation study on block-level reading order data (BLRO) and category-aware embedding (CAE).

	$\text{Text}^{\text{Edit}}\downarrow$		$\text{Order}^{\text{Edit}}\downarrow$		$\text{Overall}^{\text{Edit}}\downarrow$	
	EN	ZH	EN	ZH	EN	ZH
Baseline	0.055	0.079	0.156	0.145	0.142	0.163
+BLRO	0.045	0.078	0.066	0.123	0.119	0.158
+CAE	0.043	0.077	0.066	0.113	0.118	0.151

Table 8 Ablation study on pruning strategies.

Strategy	$\text{Text}^{\text{Edit}}\downarrow$	$\text{Formula}^{\text{Edit}}\downarrow$	$\text{Table}^{\text{TEDS}}\uparrow$	$\text{Overall}^{\text{Edit}}\downarrow$
Train from scratch	0.120	0.355	65.2	0.208
Prune less important layers [48]	0.085	0.336	78.0	0.170
Prune deeper layers [49]	0.101	0.385	78.1	0.188
Prune shallow layers	0.091	0.363	77.9	0.180
CPD	**0.067**	**0.314**	**84.4**	**0.150**

Table 9 Ablation study on CPD.

Layers	Params	$\text{Text}^{\text{Edit}}\downarrow$	$\text{Formula}^{\text{Edit}}\downarrow$	$\text{Table}^{\text{TEDS}}\uparrow$	$\text{Overall}^{\text{Edit}}\downarrow$
36	3.0	0.060	0.284	86.9	0.135
12	1.2	0.067	0.314	84.4	0.150
8	0.9	0.076	0.321	78.3	0.164
4	0.6	0.091	0.340	74.9	0.180

highlighting the efficiency advantage of the SRR paradigm.

Ablation study on block-level reading order data and category-aware embedding. The baseline model refers to LayoutReader [11], which only supports line-level relation prediction. As shown in Table 7, training on our constructed block-level reading order data improves the model's reading order performance for both Chinese and English by 5.6%. Incorporating category information into the original model further enhances reading order accuracy. Moreover, since the relation prediction model does not affect the recognition results of other tasks, it only causes a minor impact on other tasks.

Ablation study on pruning strategies. Previous studies [48,49] have explored different strategies for pruning LLMs. Shortgpt [48] tends to prune less important layers, while prior work [49] favors pruning deeper layers (11–34). In addition, we further investigate the effects of pruning shallow layers (1–24) as well as training the LLM from scratch. As shown in Table 8, our proposed CPD outperforms training from scratch by 5.8% and surpasses the other pruning strategies.

Ablation study on CPD. We conduct an ablation study to investigate the impact of the degree of parameter degradation on model performance. Specifically, we evaluate model variants retaining only 4, 8, and 12 layers of the original LLM. As shown in Table 9, retaining 12 layers results in only a marginal 1.5% overall performance drop. However, when only 4 layers are retained, text recognition performance decreases by 3.1%, while the more complex table recognition task suffers a significant 12% decline. These findings indicate that while smaller models are sufficient for simpler tasks, a larger parameter capacity is crucial for maintaining performance on complex tasks.

6 Discussion

We provide additional analysis in Figure 6, including non-rectangular text blocks (a), non-Manhattan layout (b), and paragraphs overlapping with QR code (c). As shown in Figure 6(a), non-rectangular text regions are approximated as rectangular bounding boxes during detection. In Figure 6(b), the order of images does not affect the textual semantics, allowing the model to generate the correct reading order in non-Manhattan layout. For the overlapping cases in Figure 6(c) (5, 6, 7), the model mitigates the interference by splitting the paragraph into two separate text

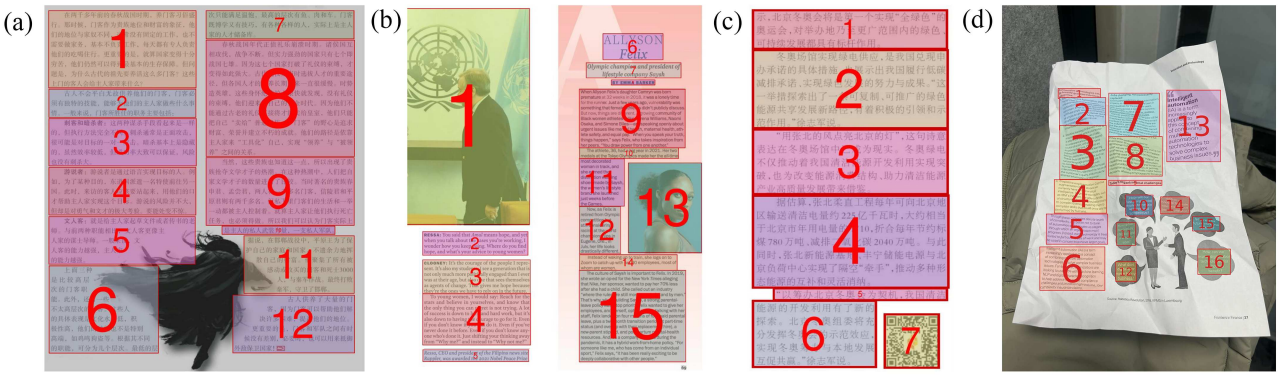


Figure 6 (Color online) Visualization results on complex layouts. (a) Non-rectangular text blocks; (b) non-Manhattan layouts; (c) text overlapping with QR codes; (d) photographed documents with geometric distortions.

blocks. Our model is not designed for photographed document scenarios. The current bounding-box-based cropping strategy assumes relatively clean layouts; in real-world photographed documents, inaccurate bounding boxes may lead to suboptimal cropping and recognition errors, as shown in Figure 6(d) (7, 8). We will address this in future work.

7 Conclusion

This paper introduces MonkeyOCR, a document parsing model based on the structure-recognition-relation triplet paradigm, which unifies structural detection, content recognition, and relation prediction 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. We further identify parameter redundancy in large multimodal models and propose contiguous parameter degradation, which accelerates inference with minimal performance drop. Experiments show that MonkeyOCR achieves state-of-the-art performance on OmniDocBench, surpassing the closed-source commercial model Gemini 2.5 Pro, demonstrating its potential as a foundation model for text-centric applications.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 62225603).

Supporting information Appendixes A–D. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1 Wang B, Xu C, Zhao X M, et al. MinerU: an open-source solution for precise document content extraction. 2024. ArXiv:2409.18839
2 Cui C, Sun T, Lin M H, et al. PaddleOCR 3.0 technical report. 2025. ArXiv:2507.05595
3 Bai S, Chen K Q, Liu X J, et al. Qwen2.5-VL technical report. 2025. ArXiv:2502.13923
4 Zheng G, Qian J Y, Tang J J, et al. Why LVLMs are more prone to hallucinations in longer responses: the role of context. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2025. 4101–4113
5 Zhou Y Y, Cui C H, Yoon J, et al. Analyzing and mitigating object hallucination in large vision-language models. In: Proceedings of the 12th International Conference on Learning Representations, 2024
6 Ouyang L K, Qu Y, Zhou H B, et al. OmniDocBench: benchmarking diverse PDF document parsing with comprehensive annotations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2025. 24838–24848
7 Livathinos N, Auer C, Lysak M, et al. Docling: an efficient open-source toolkit for AI-driven document conversion. In: Proceedings of the AAAI Workshop on Open-Source AI for Mainstream Use, 2025
8 Zhao Y A, Lv W Y, Xu S L, et al. DETRs beat YOLOs on real-time object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. 16965–16974
9 Huang Y P, Lv T C, Cui L, et al. LayoutLMv3: pre-training for document AI with unified text and image masking. In: Proceedings of the 30th ACM International Conference on Multimedia, 2022. 4083–4091
10 Wang A, Chen H, Liu L H, et al. YOLOv10: real-time end-to-end object detection. In: Proceedings of Advances in Neural Information Processing Systems, 2024. 107984–108011
11 Wang Z L, Xu Y H, Cui L, et al. LayoutReader: pre-training of text and layout for reading order detection. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021. 4735–4744

12 Li C X, Liu W W, Guo R Y, et al. PP-OCRv3: more attempts for the improvement of ultra lightweight OCR system. 2022. ArXiv:2206.03001
13 Shi B G, Bai X, Yao C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans Pattern Anal Mach Intell, 2017, 39: 2298–2304
14 Xiao K, Zhu A, Iwana B K, et al. Scene text recognition via dual character counting-aware visual and semantic modeling network. Sci China Inf Sci, 2024, 67: 139101
15 Shan G C, Wang H Y, Liang W, et al. Robust encoder-decoder learning framework for offline handwritten mathematical expression recognition based on a multi-scale deep neural network. Sci China Inf Sci, 2021, 64: 139101
16 Gu Z C, Liang G, Wang B, et al. UniMERNet: a universal network for real-world mathematical expression recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2026. 34106–34115
17 He Y L, Qi X B, Ye J Q, et al. PingAn-VCGroup's solution for ICDAR 2021 competition on scientific table image recognition to LaTeX. 2021. ArXiv:2105.01846
18 Zhao Z Y, Kang H R, Wang B, et al. DocLayout-YOLO: enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception. 2024. ArXiv:2410.12628
19 Xia R Q, Mao S, Yan X C, et al. DocGenome: an open large-scale scientific document benchmark for training and testing multi-modal large language models. 2024. ArXiv:2406.11633
20 Gu Z X, Meng C H, Wang K, et al. XYLayoutLM: towards layout-aware multimodal networks for visually-rich document understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 4583–4592
21 Kim G, Hong T, Yim M, et al. OCR-free document understanding transformer. In: Proceedings of the European Conference on Computer Vision, 2022. 498–517
22 Blecher L, Cucurull G, Scialom T, et al. Nougat: neural optical understanding for academic documents. In: Proceedings of the 12th International Conference on Learning Representations, 2024
23 Liu Y L, Chen H, Shen C H, et al. ABCNet: real-time scene text spotting with adaptive Bezier-curve network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 9809–9818
24 Peng D Z, Wang X Y, Liu Y L, et al. SPTS: single-point text spotting. In: Proceedings of the 30th ACM International Conference on Multimedia, 2022. 4272–4281
25 Liu Y L, Zhang J X, Peng D Z, et al. SPTS v2: single-point scene text spotting. IEEE Trans Pattern Anal Mach Intell, 2023, 45: 15665–15679
26 Wan J Q, Song S B, Yu W W, et al. 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, 2024. 15641–15653
27 Wei H R, Liu C L, Chen J Y, et al. General OCR theory: towards OCR-2.0 via a unified end-to-end model. 2024. ArXiv:2409.01704
28 Li Z, Yang B, Liu Q, et al. 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, 2024. 26763–26773
29 Liu Y L, Yang B, Liu Q, et al. TextMonkey: an OCR-free large multimodal model for understanding document. IEEE Trans Pattern Anal Mach Intell, 2026, 48: 6008–6019
30 Feng H, Liu Q, Liu H, et al. DocPedia: unleashing the power of large multimodal model in the frequency domain for versatile document understanding. Sci China Inf Sci, 2024, 67: 220106
31 Chen Z, Wang W Y, Tian H, et al. How far are we to GPT-4V? Closing the gap to commercial multimodal models with open-source suites. Sci China Inf Sci, 2024, 67: 220101
32 Zhu J G, Wang W Y, Chen Z, et al. InternVL3: exploring advanced training and test-time recipes for open-source multimodal models. 2025. ArXiv:2504.10479
33 Poznanski J, Rangapur A, Borchardt J, et al. olmOCR: unlocking trillions of tokens in PDFs with vision language models. 2025. ArXiv:2502.18443
34 Wei H R, Sun Y F, Li Y K. DeepSeek-OCR: contexts optical compression. 2025. ArXiv:2510.18234
35 Du Y K, Chen Z N, Xie Y Z, et al. UniRec-0.1B: unified text and formula recognition with 0.1B parameters. In: Proceedings of the European Conference on Computer Vision, 2026. 36–55
36 Cheng H Y, Zhang P R, Wu S H, et al. 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, 2023. 15138–15147
37 Da C, Luo C, Zheng Q, et al. Vision grid transformer for document layout analysis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023. 19462–19472
38 Pfitzmann B, Auer C, Dolfi M, et al. DocLayNet: a large human-annotated dataset for document-layout segmentation. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022. 3743–3751
39 Zheng X Y, Burdick D, Popa L, et al. 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, 2021. 697–706
40 Zhong X, ShafieiBavani E, Jimeno Yepes A. Image-based table recognition: data, model, and evaluation. In: Proceedings of the European Conference on Computer Vision, 2020. 564–580
41 Yuan Y, Liu X, Dikubab W, et al. Syntax-aware network for handwritten mathematical expression recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 4553–4562
42 Liu Y L, Li Z, Huang M X, et al. OCRBench: on the hidden mystery of OCR in large multimodal models. Sci China Inf Sci, 2024, 67: 220102
43 Comanici G, Bieber E, Schaekermann M, et al. Gemini 2.5: pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. 2025. ArXiv:2507.06261
44 Sun T, Cui C, Du Y N, et al. PP-DocLayout: a unified document layout detection model to accelerate large-scale data construction. 2025. ArXiv:2503.17213
45 Hurst A, Lerer A, Goucher A, et al. GPT-4o system card. 2024. ArXiv:2410.21276
46 Nassar A, Omenetti M, Lysak M, et al. Smoldocling: an ultra-compact vision-language model for end-to-end multi-modal document conversion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2025. 21972–21983
47 Feng H, Wei S, Fei X, et al. Dolphin: document image parsing via heterogeneous anchor prompting. In: Proceedings of Findings of the Association for Computational Linguistics: ACL 2025, 2025. 21919–21936
48 Men X, Xu M Y, Zhang Q Y, et al. ShortGPT: layers in large language models are more redundant than you expect. In: Proceedings of Findings of the Association for Computational Linguistics: ACL 2025, 2025. 20192–20204
49 Gromov A, Tirumala K, Shapourian H, et al. The unreasonable ineffectiveness of the deeper layers. In: Proceedings of the 13th International Conference on Learning Representations, 2025