KITAB-Bench: A Comprehensive Multi-Domain Benchmark for Arabic OCR and Document Understanding

Anonymous ACL submission

Abstract

With the growing adoption of Retrieval-Augmented Generation (RAG) in document processing, robust text recognition has become increasingly critical for knowledge extraction. While OCR (Optical Character Recognition) for English and other languages benefits from large datasets and well-established benchmarks, Arabic OCR faces unique challenges due to its cursive script, right-to-left text flow, and complex typographic and calligraphic features. We present **`KITAB-Bench`**, a comprehensive Arabic OCR benchmark that fills the gaps in current evaluation systems. Our benchmark comprises 8,809 samples across 9 major domains and 36 sub-domains, encompassing diverse document types including handwritten text, structured tables, and specialized coverage of 21 chart types for business intelligence. Our findings show that modern vision-language models (such as GPT-4, Gemini, and Qwen) outperform traditional OCR approached (like EasyOCR, PaddleOCR, and Surya) by an average of 60% in Character Error Rate (CER). Furthermore, we highlight significant limitations of current Arabic OCR models, particularly in PDF-to-Markdown conversion, where the best model Gemini-2.0-Flash achieves only 65% accuracy. This underscores the challenges in accurately recognizing Arabic text, including issues with complex fonts, numeral recognition errors, word elongation, and table structure detection. This work establishes a rigorous evaluation framework that can drive improvements in Arabic document analysis methods and bridge the performance gap with English OCR technologies.

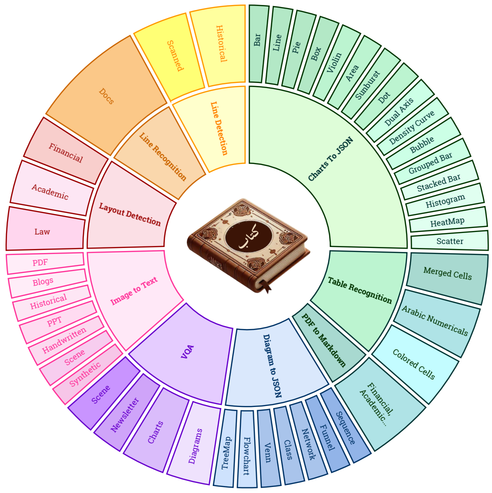

Figure 1: Overview of the core domains and sub-domains in KITAB-Bench. Our benchmark spans nine major domains (e.g., OCR, charts to JSON, table recognition) and 36 sub-domains (e.g., scanned text, handwritten text, various chart types), providing a comprehensive evaluation framework for modern Arabic document processing and analysis.

1 Introduction

With the upsurge in adoption of Retrieval-Augmented Generation (RAG) based systems for document processing, the quality of document ingestion pipelines has become increasingly critical. Optical Character Recognition (OCR) plays a crucial role in this pipeline, enabling the conversion of physical documents into machine-readable text and databases for enabling effective knowledge retrieval. Although significant progress has been made in the multilingual OCR (JaidedAI, 2020; Fu et al., 2024; Wei et al., 2024; Smith, 2007), with comprehensive datasets like PubLayNet (Zhong et al., 2019b), DocBank (Li et al., 2020), M6Doc (Cheng et al., 2023), and DocLayNet (Pfitzmann et al., 2022), Arabic OCR continues to lag behind. This gap is largely due to the unique challenges of the Arabic script, including its cursive nature, complex typography, and right-to-left text orientation.

Existing Arabic OCR datasets (Table 1), like KHATT (Mahmoud et al., 2014) and IFN/ENIT (Pechwitz et al., 2002) focus mainly on handwritten text, whereas APTI (Slimane et al., 2009) covers only specific aspects of printed text. These efforts fail to address advanced document

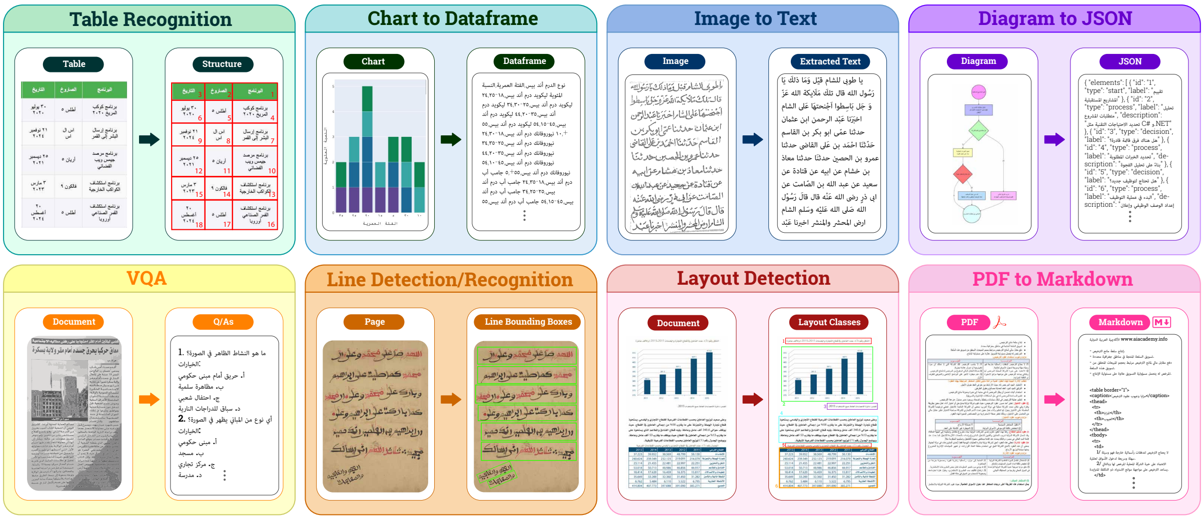

Figure 2: Overview of different tasks in our benchmark: Eight key components illustrating the task inputs and outputs for table recognition, chart understanding, text recognition, diagram analysis, VQA, line detection, layout analysis, and PDF-to-Markdown conversion, complete with input/output examples for each task.

Domain/ Characteristics	EXAMS-V*	Camel-Bench	MIDAD†	KHATT	KITAB-Bench (Ours)
PDF to Markdown	✗	✗	✗	✗	✓
Layout Detection	✗	✗	✗	✗	✓
Line Detection	✗	✗	✗	✗	✓
Line Recognition	✗	✓	✗	✗	✓
Table Recognition	✗	✗	✗	✗	✓
Image to Text	✓	✓	✓	✓	✓
Charts to JSON	✗	✗	✗	✗	✓
Diagram to Code	✗	✗	✗	✗	✓
VQA	✓	✓	✗	✗	✓
Handwritten Samples	✗	✗	✓	✓	✓
Open Source	✓	✓	✗	✓	✓
Total Samples (#)	823	3,004	29,435	5,000	8,809

Table 1: Comparison of Arabic OCR Benchmarks Across Different Domains. Benchmarks compared: LaraBench (Abdelali et al., 2023), CamelBench (Ghaboura et al., 2024), MIDAD (Bhatia et al., 2024), KHATT (Mahmoud et al., 2014), and KITAB-Bench (Ours). (∗: Only the Arabic samples are considered.) (†: The test set of the dataset is considered.)

processing challenges such as table parsing, font detection, and numeral recognition. Arabic benchmarks like CAMEL-Bench (Ghaboura et al., 2024) and LAraBench (Abdelali et al., 2023) evaluate large multimodal and language models, but they give limited attention to document understanding tasks. Consequently, there remains a need for a more comprehensive framework to systematically evaluate and compare Arabic OCR solutions. Our benchmark addresses these gaps by offering diverse document types and evaluation tasks to facilitate in-depth assessments of modern OCR systems.

We present KITAB-Bench, a comprehensive Arabic OCR benchmark spanning 9 domains and 36 sub-domains. Our framework evaluates layout detection (text blocks, tables, figures), multi-format recognition (printed/handwritten text, charts, diagrams), and structured output generation (HTML tables, DataFrame charts, markdown). This enables rigorous assessment of both basic OCR capabilities and advanced document understanding tasks.

The contributions of this work include (1) A comprehensive Arabic OCR benchmark covering multiple document types and recognition tasks. (2) Detailed evaluation metrics for assessing performance across different document understanding challenges. We also propose CharTeX and CODM metric to evaluate chart extraction and diagram extraction respectively. (3) Baseline results for popular OCR systems and Vision Language Models (VLMs), highlighting current limitations and areas for improvement. (4) A standardized framework for comparing Arabic OCR systems, facilitating future research and development.

2 Related Work

The development of robust Optical Character Recognition (OCR) systems has been extensively studied across document layout analysis (Zhao et al., 2024; Shen et al., 2021; Paruchuri, 2024b; JaidedAI, 2020; Auer et al., 2024; Li et al., 2020), table detection (Li et al., 2019; Paliwal et al., 2019; Nassar et al., 2022; Li et al., 2021; Schreiber et al., 2017), and document understanding (Staar et al., 2018; Weber et al., 2023; Livathinos et al., 2021). While English OCR benefits from rich datasets like PubLayNet (Zhong et al., 2019b), DocBank (Li et al., 2020), M6Doc (Cheng et al.,

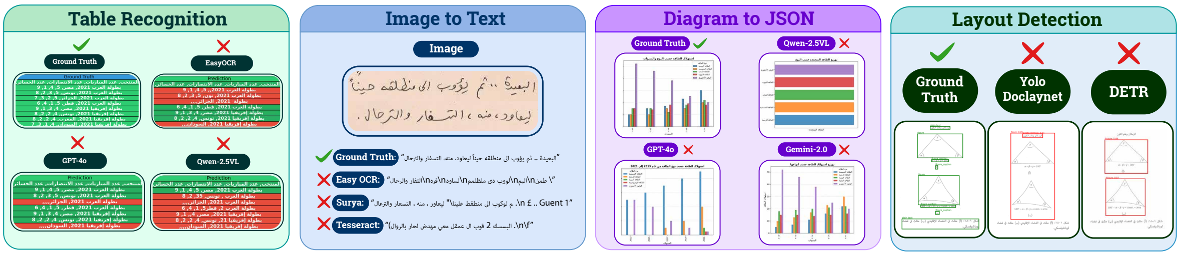

Figure 3: Comparison of model performance across four document understanding tasks (Table Recognition, Image to Text, Diagram to JSON, and Layout Detection) showing successful and failed cases for different models including Ground Truth, EasyOCR, GPT-4, Qwen, Surya, Tesseract, Yolo, and DETR on Arabic document benchmark data.

Domain	**Total Samples**
PDF to Markdown	33
Layout	2,100
Line Detection	378
Line Recognition	378
Table Recognition	456
Image to Text	3,760
Charts to DataFrame	576
Diagram to Json	226
VQA	902
Total	**8,809**

Table 2: Distribution of samples across different domains in our dataset. A more detailed count for different sub-domains and data sources is in Appendix A.

2023), and DocLayNet (Pfitzmann et al., 2022), Arabic lacks standardized benchmarks for diverse fonts and layouts. Recent efforts like MIDAD (Bhatia et al., 2024) curates extensive training data for Arabic OCR and handwriting recognition, while Peacock (Alwajih et al., 2024) introduces culturally-aware Arabic multimodal models. Existing resources such as CAMEL-Bench (Ghaboura et al., 2024), LAraBench (Abdelali et al., 2023), MADAR (Bouamor et al., 2018), OSACT (Mubarak et al., 2022), and Tashkeela (Zerrouki and Balla, 2017) focus on language modeling or specific tasks rather than full-page OCR evaluation. Handwriting datasets including HistoryAr (Pantke et al., 2014), IFN/ENIT (Pechwitz et al., 2002), KHATT (Mahmoud et al., 2014), APTI (Slimane et al., 2009), and Muharaf (Saeed et al., 2024) emphasize word/line recognition over document structure analysis.

Arabic table recognition faces challenges from merged cells and RTL formatting (Pantke et al., 2014). While methods like GTE (Zheng et al., 2021), GFTE (Li et al., 2021), CascadeTabNet (Prasad et al., 2020), TableNet (Paliwal et al., 2019), and TableFormer (Nassar et al., 2022) advance Latin table detection, their effectiveness on Arabic documents remains unproven. Document conversion pipelines (CCS (Staar et al., 2018), Tesseract (Smith, 2007), Docling (Auer et al., 2024), Surya (Paruchuri, 2024b), Marker (Paruchuri, 2024a), MinerU (Wang et al., 2024a), PaddleOCR (Du et al., 2020)) lack Arabic-specific optimizations for segmentation and diacritic handling (Mahmoud et al., 2018; Kiessling et al., 2019). This highlights the critical need for comprehensive Arabic OCR benchmarks addressing text recognition, table detection, and layout parsing.

3 KITAB-Bench

Our methodology offers a novel approach to benchmarking Arabic OCR systems via a comprehensive data collection strategy and a systematic evaluation framework. We gather curated samples from existing Arabic document datasets, manually collected and annotated PDFs, and employ a five-phase LLM-assisted human-in-the-loop pipeline (Figure 4) to generate diverse supplementary content. Our evaluation framework spans nine specialized tasks, enabling thorough assessment of OCR performance across various document processing challenges and providing a robust benchmark for Arabic document understanding tasks.

3.1 PDF Data Collection

We curated 33 diverse PDFs from online sources in academia, medicine, law, and literature. To ensure challenging cases, we selected documents featuring richly formatted tables with extensive color usage, merged cells, Arabic numerals, historical texts, watermarks, and handwritten annotations. Each PDF averaged three pages, and we then manually annotated them. This dataset comprehensively captures real-world complexities, making it a valuable benchmark for PDF-to-Markdown conversion.

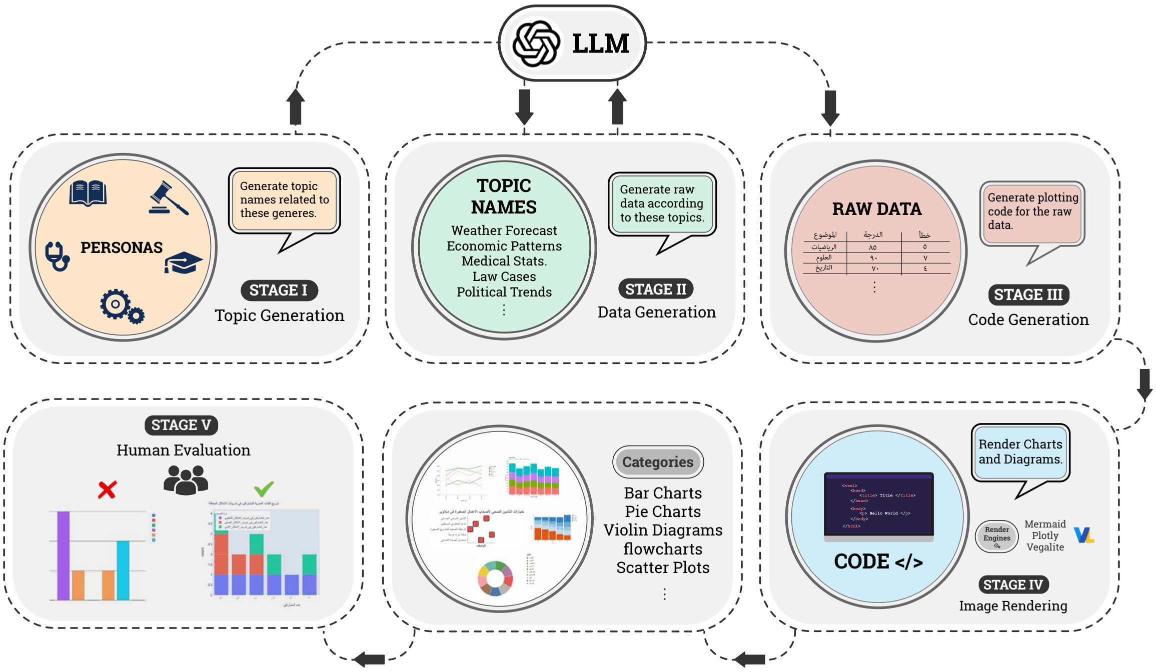

Figure 4: Synthetic Data Generation Pipeline: A 5-stage process using LLMs to generate topics, create raw data, produce visualization code, render charts, and perform human evaluation for quality control.

3.2 LLM-Assisted Data Generation Pipeline

To generate data for charts, diagrams and tables, we implemented a five-phase LLM-assisted generation pipeline with human validation at critical stages, as illustrated in Figure 4. *In Phase I (Topic Generation)*, our system employs an LLM to generate diverse topic names across multiple domains. This phase incorporates various personas (academic, legal, medical, technical) to ensure broad coverage of document types. *Phase II (Data Generation)* transforms the validated topics into structured raw data. The LLM generates content following Arabic linguistic and formatting conventions across various domains. *In Phase III (Code Generation)*, the system converts the validated raw data into plotting code, with special attention to Arabic text rendering requirements and RTL content management. *Phase IV (Image Rendering)* utilizes specialized rendering engines (Mermaid, Plotly, Vegalite, HTML) to create visual representations while maintaining Arabic text integrity.

The final phase (Human Evaluation) implements rigorous quality control through expert validation. Evaluators filter charts, tables and diagrams based on detected anomalies and ensure adherence to Arabic-specific document conventions. This phase is crucial for maintaining the high quality of our benchmark dataset.

3.3 Dataset Statistics

Our benchmark dataset comprises over 8,809 samples across 9 major domains and 36 sub-domains, representing a comprehensive collection of Arabic document types for OCR evaluation. As detailed in Table 8, the dataset combines carefully curated samples from established datasets, manually annotation PDFs, and synthetically generated content created through our LLM-assisted pipeline (Figure 4). The Image-to-Text portion (3,760 samples) includes data from historical documents (HistoryAr (Pantke et al., 2014)), handwritten text collections (Khatt (Mahmoud et al., 2014), ADAB (Boubaker et al., 2021), Muharaf (Saeed et al., 2024)), and scene text (EvAREST (Hassan et al., 2021)), while layout detection comprises 2,100 samples from BCE-Arabic-v1 (Saad et al., 2016) and DocLayNet (Pfitzmann et al., 2022).

For layout analysis, we incorporated 1,700 samples from BCE-Arabic-v1 dataset (Saad et al., 2016), 400 samples from DocLayNet dataset (Pfitzmann et al., 2022) focusing on financial, academic, legal, and patent documents. The line detection and recognition tasks contains 378 samples each from self-developed dataset. We further enriched the dataset with 500 samples from PATS-A01 (El-Muhtaseb, 2010) benchmark to ensure diverse representation. For handwritten text recognition, we assembled a comprehensive collection of 1,000

Task	Metric	Surya	Tesseract	EasyOCR
Detection	mAP@50	**79.67**	46.39	68.02
	mAP@0.5:0.95	27.40	14.30	**32.74**
Recognition	WER	1.01	1.00	**0.53**
	CER	0.87	0.66	**0.20**

Table 3: Performance of different models on Line Detection and Line Recognition Task on our Benchmark

samples combining datasets from Khatt (Mahmoud et al., 2014) (both paragraph and line-level annotations), Adab (Boubaker et al., 2021), Muharaf (Saeed et al., 2024), and OnlineKhatt (Mahmoud et al., 2018). The benchmark also includes specialized content from ISI-PPT (Wu and Natarajan, 2017) (500 samples), and Hindawi (Elfilali, 2023) (200 samples) for various document types. Scene text understanding is supported by 800 samples from EvArest (Hassan et al., 2021), providing real-world context diversity. A detailed table showing all the dataset is provided in the Appendix A.

A significant portion of our dataset consists of synthetically generated content, including 576 samples for Charts-to-DataFrame (spanning 16 different chart types), 422 samples for Diagram-to-Code (covering sequence diagrams, flowcharts, and tree maps), 456 samples for Tables-to-CSV/HTML, and 902 samples for VQA tasks. These synthetic samples were generated through our five-phase LLM-assisted human-in-the-loop pipeline (Figure 4). Every sample in our dataset - whether from existing sources or newly generated - underwent validation by native Arabic speakers before inclusion in the final benchmark. This rigorous validation, reinforced by expert review and automated checks, ensures high quality and authenticity across all domains. A detailed analysis is in Appendix C.

4 Experiments

Our experimental evaluation comprehensively assesses the capabilities of current OCR systems and state-of-the-art vision-language models (VLMs) across different Arabic and multilingual document understanding tasks. Figure 2 illustrates the nine distinct tasks in our evaluation framework.

We evaluate three categories of systems: VLMs, traditional OCR systems, and specialized document processing tools. For VLMs, we include both closed-source models like `gpt-4o-2024-08-06`, `gpt-4o-mini-2024-07-18` (Hurst et al., 2024; Achiam et al., 2023), and `gemini-2.0-flash` (Georgiev et al., 2024; Google DeepMind, 2025), as well as open-source alternatives such as `Qwen2-VL-7B` (Wang et al., 2024b), `Qwen2.5-VL-7B` (Team, 2025), and the `AIN-7B` (Heakl et al., 2025). Traditional OCR approaches in our evaluation include `Surya` (Paruchuri, 2024b), `Tesseract` (Smith, 2007), `EasyOCR` (JaidedAI, 2020), and `PaddleOCR` (Li et al., 2022; Du et al., 2021). For specialized document processing tasks, we employ systems like `Docling` (Auer et al., 2024), and `Marker` (Paruchuri, 2024a). Layout detection capabilities are evaluated using methods implemented in `Surya-layout` (Paruchuri, 2024b), Yolo-doclaynet (Zhao et al., 2024) from `MinerU` (Wang et al., 2024a), and `RT-DETR` (Zhao et al., 2023) based method in Docling (Auer et al., 2024).

4.1 Evaluation Frameworks and Metrics

Our evaluation framework comprises nine specialized tasks designed to assess different aspects of Arabic OCR systems, as demonstrated in Figure 2. Each task addresses specific challenges in Arabic document processing. For this reason, we employ task-specific metrics to evaluate different aspects of document understanding.

PDF-to-Markdown: It evaluates the conversion of Arabic PDFs to structured markdown while preserving the text and table structure. Since both table and text structure are important, for evaluating PDF to Markdown conversion quality, we propose MARS (Markdown Recognition Score), which combines chrF (Popović, 2015) with Tree-Edit-Distance-based Similarity (TEDS) (Zhong et al., 2020) :

$$\mathrm{MARS} = \alpha \cdot \mathrm{chrF}_3 + (1-\alpha) \cdot \mathrm{TEDS}(T_a, T_b) \quad (1)$$

where α $(0 \leq \alpha \leq 1)$ is the weight. T_a represent

Dataset	Metric	Surya	Yolo-doclaynet	Detr (docling)
BCE	mAP@0.5	0.506	0.470	**0.750**
	mAP@0.5:0.95	0.381	0.369	**0.566**
	Precision	**0.751**	0.608	0.626
	Recall	0.593	0.592	**0.725**
	F1 Score	0.635	0.585	**0.654**
DocLayNet	mAP@0.5	0.675	0.404	**0.758**
	mAP@0.5:0.95	0.469	0.335	**0.541**
	Precision	**0.782**	0.527	0.635
	Recall	0.856	0.503	**0.770**
	F1 Score	0.799	0.499	**0.670**

Table 4: Performance comparison of layout detection models using different evaluation metrics

predicted table structure and T_b the ground truth structure.

Table Recognition: We evaluate table extraction using both HTML and CSV formats, where HTML format (evaluated using TEDS (Zhong et al., 2020)) preserves rich structural information including cell spans and hierarchical relationships crucial for complex Arabic tables, while CSV format (evaluated using Jaccard Index 2) focuses on raw data extraction optimized for machine processing and data analysis pipelines. This dual-format evaluation ensures systems can both maintain complex table structures for human readability and provide clean, structured data for automated processing, specifically important for RAG based systems.

$$J(P,G) = \frac{|P \cap G|}{|P \cup G|} = \frac{|P \cap G|}{|P| + |G| - |P \cap G|} \quad (2)$$

where $|P \cap G|$ represents the number of exact matching cells between predicted and ground truth tables, and $|P \cup G|$ represents the total number of unique cells across both tables.

Chart-to-Dataframe: This task evaluates extracting structured data from Arabic charts into machine-readable dataframes. Systems must accurately parse numerical values, text labels, and preserve data relationships across chart types (bar, line, pie). We use the Structuring Chart-oriented Representation Metric (SCRM) (Xia et al., 2024)—which combines type recognition, topic understanding, and structural numerical fidelity (see Appendix D)—and also propose our own CharTeX (Chart Extraction Score) metric. CharTeX combines the cHrf scores for chart type and topic with the jaccord index for the dataframe, using fuzzy matching (80% threshold) when columns do not exactly align.

$$\text{Metric} = \alpha J_{type} + \beta J_{topic} + (1 - \alpha - \beta) J_{data} \quad (3)$$

Here, J_{type} and J_{topic} denote the chrF scores between the predicted and ground-truth chart type and topic, while J_{data} measures the structural similarity of the predicted and ground-truth JSON data.

Diagram-to-JSON: This task evaluates the conversion of Arabic flowcharts and technical diagrams into JSON while preserving semantic relationships and technical specifications. We propose CODM (Code-Oriented Diagram Metric), extending SCRM (Xia et al., 2024), with the same fomulation as in Eq 3. More detail about this metric is provided in Appendix E

Image-to-Text: This task assess the basic text recognition capabilities across different Arabic fonts and styles, including the handling of cursive script connections, diacritical marks, and various text orientations. We use we use Character Error Rate (CER) and Word Error Rate (WER). For a predicted text sequence $\hat{y}$ and ground truth sequence y, CER is computed as: CER $= \frac{\mathrm{L}(y,\hat{y})}{|y|}$, where $\mathrm{L}(y, \hat{y})$ is the Levenshtein distance between character sequences and $|y|$ is the ground truth length. WER is calculated the same way with words as the unit of error.

Visual Question Answering: Tests the ability of models to understand and reason about Arabic document content, we evaluate using standard accuracy for MCQ questions and exact word match.

Line Detection: Focuses on the accurate identification and processing of individual text lines in Arabic documents. We evaluate using mean Average Precision (mAP) at different Intersection over Union (IoU) thresholds: mAP@0.5 and mAP@0.5:0.95, which assess the localization accuracy of detected text lines.

Layout Detection: Assesses document structure analysis capabilities, including the identification of headers, paragraphs, and complex layout elements in Arabic documents. Performance is measured using mAP@0.5 and mAP@0.5:0.95 for localization accuracy, complemented by Precision, Recall, and F1 scores to evaluate the overall detection quality across different layout components.

All metrics are computed on our diverse benchmark dataset, which encompasses various document types and complexity levels in both Arabic and multilingual contexts. Table 10 provides a detailed mapping of tasks, metrics, and evaluated systems.

4.2 Experimental Setup

We implement our evaluation pipeline with careful consideration of hyperparameters for different metric. All experiments use NVIDIA A100 GPUs. For VLMs, we use their official implementations or API endpoints. Traditional OCR systems are evaluated using pre-trained models provided by the frameworks. For PDF-to-Markdown evaluation metric MARS 1, we choose $\alpha = 0.5$ and $\alpha = 0.5$ and $\beta = 0.2$ for Diagram-to-JSON evaluation metric CODM. We average the results over multiple runs, with performance comparisons shown in different tables [3, 6, 5, 7, and 4].

Model Group	Models	Table Extraction		End-to-End PDF		
		TEDS (HTML)	Jaccard (CSV)	CHrF (Text)	TEDS (Table)	MARS
Closed	GPT-4o	**85.76**	**66.36**	69.62	**60.61**	65.12
	GPT-4o-mini	69.32	49.50	56.59	52.69	54.64
	Gemini-2.0-Flash	83.08	65.55	**75.75**	55.55	**65.65**
Open	Qwen2-VL-7B	57.83	40.20	40.30	2.54	21.42
	Qwen2.5-VL-7B	59.31	59.58	69.21	11.65	40.43
	AIN-7B	75.94	64.83	56.52	49.32	52.92
Framework	Tesseract	28.23^D 38.64^I	14.85^D 16.04^I	59.91^D	45.44^D	52.68^D
	EasyOCR	49.10^D 39.09^I	23.83^D 17.88^I	57.46^D	51.12^D	54.29^D
	Surya	50.15^M	70.42^M	58.38^M	44.29^M	51.34^M

^{D}Docling (Auer et al., 2024) pipeline ^{I}Img2Table (Cattan, 2021) pipeline ^{M}Marker (Paruchuri, 2024a) pipeline

Table 5: Performance comparison of different models for table extraction and end-to-end PDF to markdown conversion tasks on our benchmark.

Group	Models	CHrF ↑	CER ↓	WER ↓
Closed	GPT-4o	61.01	0.31	0.55
	GPT-4o-mini	47.21	0.43	0.71
	Gemini-2.0-Flash	77.95	**0.13**	0.32
Open	Qwen2VL-7B	33.94	1.48	1.55
	Qwen2.5VL-7B	49.23	1.20	1.41
	AIN-7B	**78.33**	0.20	**0.28**
Framework	Tesseract	39.62	0.54	0.84
	EasyOCR	45.47	0.58	0.89
	Paddle	16.73	0.79	1.02
	Surya	20.61	4.95	5.61

Table 6: Performance comparison of models for OCR (image to text) tasks on our benchmark. A detailed performance comparison among different open-source dataset is available in Appendix B

5 Results and Discussion

In this section, we present a comprehensive evaluation of different models across different tasks of our framework. The results provide a clear distinction between the performance of closed-source models, open-source models, and framework-based solutions, revealing both their strengths and limitations. We observe very clear performance gap between closed and open-source solutions. While closed-source models like Gemini-2.0-Flash consistently outperform other models almost all the tasks.

5.1 Charts, Diagrams, and VQA

Table [7] presents model performance across different chart and diagram understanding tasks, evaluated using SCRM and CharTeX (for charts), and VQA-based accuracy metrics. Among closed-source models, Gemini-2.0 achieves the highest performance on chart understanding metrics, scoring 71.4% on SCRM and 56.28% on CharTeX. The performance gap between Gemini-2.0 and GPT-4o is particularly pronounced in CharTeX evaluation (10.33%) compared to SCRM (2.8%). Open-source models shows a significant limitation in complex chart understanding. While their SCRM scores remain competitive, both Qwen variants score below 23% on CharTeX evaluation. The visual question-answering results reveal an important exception to the general closed-source advantage. AIN achieves 87% on PATDVQA, surpassing Gemini-2.0 by 11.5%. AIN also shows competitive performance on MTVQA (31.50%), which is similar to GPT-4o and 4% better than GPT-4o-mini. This shows that open-source models can be competitive with closed-source alternatives.

5.2 Layout and Lines: Document Structure

Our evaluation of document structure understanding reveals distinct performance patterns across layout detection and line processing tasks. In layout detection (Table 4), RT-DETR (Zhao et al., 2023) achieves superior overall performance with mAP0.5 scores of 0.750 and 0.758 on BCE (arabic only) and DocLayNet (english) datset respectively. However, Surya (Paruchuri, 2024b) demonstrates higher precision (0.782 on DocLayNet, 0.751 on BCE), despite lower recall rates. This trade-off suggests that different architectures optimize for different aspects of layout detection.
The line processing results (Table 3) highlight a clear contrast between detection and recognition capabilities. While Surya excels in detection with a mAP@0.50 of 79.67%, EasyOCR demonstrates superior recognition performance (WER: 0.53, CER:

Group	Model	Chart		Diagram	Visual QA				
		SCRM	CharTeX	CODM	MTVQA^O	ChartsVQA^M	DiagramsVQA^M	PATDVQA^M	Average
Closed	GPT-4o	68.6	45.95	61.6	32.00	77.00	85.29	82.50	69.19
	GPT-4o-mini	67.2	43.33	61.4	26.80	58.00	83.33	80.00	62.03
	Gemini-2.0-Flash	**71.4**	**56.28**	**71.8**	**35.00**	72.00	**88.24**	75.50	67.68
Open	Qwen2-VL-7B	56.6	21.59	63.0	19.60	59.00	82.35	77.50	59.61
	Qwen2.5-VL-7B	36.2	22.08	59.2	23.00	74.00	79.41	74.50	62.72
	AIN-7B	66.6	34.61	66.40	31.50	75.00	85.29	**87.00**	**69.69**

Table 7: Model Performance on Chart Understanding, Diagram Parsing, and Visual Question Answering Tasks. For VQA tasks, O denotes open-ended question type from MTVQA (Tang et al., 2024) dataset and M denotes MCQ type questions.

0.20). This inverse relationship between detection and recognition performance across models indicates a fundamental challenge in optimizing both capabilities simultaneously. Notably, Tesseract shows consistent but lower performance across both metrics, suggesting that newer architectures have made significant improvements over traditional approaches. We also observe that no single model excels at both detection and recognition, which requires for hybrid solutions.

5.3 Tables, OCR, and PDF-to-Markdown

Across table extraction tasks (Table 5), closed-source models maintain a clear advantage, with GPT-4o achieving 85.76% TEDS and 66.36% Jaccard scores. Among open-source models, AIN (75.94% TEDS) significantly outperforms Qwen variants, while specialized frameworks like Surya achieve competitive results (70.42% Jaccard) through targeted pipelines. In OCR evaluation (Table 6), Gemini-2.0-Flash leads with the lowest error rates (CER: 0.13, WER: 0.32). Notably, AIN matches this performance level (WER: 0.28), while traditional OCR frameworks like EasyOCR and Tesseract show moderate performance (CER: 0.58, 0.54). The significant performance drop in Paddle (CER: 0.79) and Surya (CER: 4.95) highlights the challenges in developing robust OCR systems.

End-to-end document processing (Table 5) reveals the largest gaps between approaches. Closed-source models maintain consistent performance (GPT-4o: 65.12% MARS, Gemini-2.0: 65.65% MARS), while open-source models show substantial degradation (Qwen2-VL-7B: 21.42% MARS). Framework approaches achieve better stability, with Tesseract and EasyOCR scoring above 50% MARS, suggesting that specialized pipelines can partially bridge the gap with larger models in complete document processing tasks.

Our comprehensive evaluation demonstrates that while closed-source models maintain superior performance over open-source models across most tasks, specialized frameworks like Surya, RT-DETR Layout, and EasyOCR achieve competitive performance in targeted scenarios like table extraction, layout detection, and text recognition respectively. However, this framework advantage significantly diminishes in end-to-end pdf-to-markdown tasks where the integration capabilities of large models prove crucial, as evidenced by the performance gaps between commercial VLMs and traditional systems like EasyOCR, Surya and Tesseract in End-to-End PDF task (Table 5).

6 Conclusion

We introduce a comprehensive benchmark for Arabic OCR that fills the gap in standardized evaluation frameworks for Arabic document processing. Our dataset of 8,809 samples across nine major domains is the most diverse collection assembled for OCR evaluation, incorporating handwritten, scanned, synthetic, and scene text, as well as complex tables, charts, and end-to-end pdf-to-markdown. This framework extends beyond simple text recognition to include structural document analysis and enables systematic assessment of OCR performance across various fonts, styles, and layouts.

7 Limitations and Future Directions

Despite its contributions, this benchmark has limitations. While it covers diverse Arabic document types, it lacks full representation of historical manuscripts and low-resource dialects. Future work should expand to include these, along with scanned records from government, academic, and financial institutions.

Another key limitation is in table and chart recognition, where OCR models struggle with structure preservation, header detection, and merged cell parsing. Though our benchmark introduces challenges in these areas, further refinements are needed for robust multimodal OCR capable of jointly processing text, tables, and figures. Future advancements should focus on dataset expansion, novel evaluation metrics, deep learning refinements, and cross-lingual OCR innovations to enhance Arabic VLMs.

References

Ahmed Abdelali, Hamdy Mubarak, Shammur Absar Chowdhury, Maram Hasanain, Basel Mousi, Sabri Boughorbel, Yassine El Kheir, Daniel Izham, Fahim Dalvi, Majd Hawasly, et al. 2023. Larabench: Benchmarking arabic ai with large language models. *arXiv preprint arXiv:2305.14982*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Fakhraddin Alwajih, El Moatez Billah Nagoudi, Gagan Bhatia, Abdelrahman Mohamed, and Muhammad Abdul-Mageed. 2024. Peacock: A family of arabic multimodal large language models and benchmarks. *arXiv preprint arXiv:2403.01031*.

Christoph Auer, Maksym Lysak, Ahmed Nassar, Michele Dolfi, Nikolaos Livathinos, Panos Vagenas, Cesar Berrospi Ramis, Matteo Omenetti, Fabian Lindlbauer, Kasper Dinkla, et al. 2024. Docling technical report. *arXiv preprint arXiv:2408.09869*.

Gagan Bhatia, El Moatez Billah Nagoudi, Fakhraddin Alwajih, and Muhammad Abdul-Mageed. 2024. Qalam: A multimodal llm for arabic optical character and handwriting recognition. *arXiv preprint arXiv:2407.13559*.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, et al. 2018. The madar arabic dialect corpus and lexicon. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.

Houcine Boubaker, Abdelkarim Elbaati, Najiba Tagougui, Haikal El Abed, Monji Kherallah, Volker Märgner, and Adel M. Alimi. 2021. Adab database.

Hassina Bouressace and Janos Csirik. 2019. Printed arabic text database for automatic recognition systems. In *Proceedings of the 2019 5th International Conference on Computer and Technology Applications*, pages 107–111.

Xavier Cattan. 2021. img2table: Extract tables from images and scanned pdfs. https://github.com/xavctn/img2table. Accessed: 2025-02-14.

H. Cheng, P. Zhang, S. Wu, et al. 2023. 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 (CVPR)*.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. 2024. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*.

Yuning Du, Chenxia Li, Ruoyu Guo, Cheng Cui, Weiwei Liu, Jun Zhou, Bin Lu, Yehua Yang, Qiwen Liu, Xiaoguang Hu, et al. 2021. Pp-ocrv2: Bag of tricks for ultra lightweight ocr system. *arXiv preprint arXiv:2109.03144*.

Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, et al. 2020. Pp-ocr: A practical ultra lightweight ocr system. *arXiv preprint arXiv:2009.09941*.

Husni A. El-Muhtaseb. 2010. Pats-a01 - an arabic text database. https://faculty.kfupm.edu.sa/ics/muhtaseb/ArabicOCR/PATS-A01.htm. Database for Arabic Text Recognition Research.

Ali Elfilali. 2023. Hindawi books dataset. https://huggingface.co/datasets/Ali-C137/Hindawi-Books-dataset. Dataset.

Ling Fu, Biao Yang, Zhebin Kuang, Jiajun Song, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, Mingxin Huang, et al. 2024. Ocrbench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. *arXiv preprint arXiv:2501.00321*.

Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Sara Ghaboura, Ahmed Heakl, Omkar Thawakar, Ali Alharthi, Ines Riahi, Abduljalil Saif, Jorma Laaksonen, Fahad S Khan, Salman Khan, and Rao M Anwer. 2024. Camel-bench: A comprehensive arabic lmm benchmark. *arXiv preprint arXiv:2410.18976*.

Google DeepMind. 2025. Gemini Model Updates - February 2025. Accessed: 2025-02-14.

Heba Hassan, Ahmed El-Mahdy, and Mohamed E Hussein. 2021. Arabic scene text recognition in the deep learning era: Analysis on a novel dataset. *IEEE Access*, 9:107046–107058.

Ahmed Heakl, Sara Ghaboura, Omkar Thawkar, Fahad Shahbaz Khan, Hisham Cholakkal, Rao Muhammad Anwer, and Salman Khan. 2025. Ain: The arabic inclusive large multimodal model. *arXiv preprint arXiv:2502.00094*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

JaidedAI. 2020. Easyocr: Ready-to-use optical character recognition with multi-language support. https://github.com/JaidedAI/EasyOCR. Accessed: 2025-02-14.

Benjamin Kiessling, Daniel Stökl Ben Ezra, and Matthew Thomas Miller. 2019. Badam: A public dataset for baseline detection in arabic-script manuscripts. In *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*, page 13–18, New York, NY, USA. Association for Computing Machinery.

Chenxia Li, Weiwei Liu, Ruoyu Guo, Xiaoting Yin, Kaitao Jiang, Yongkun Du, Yuning Du, Lingfeng Zhu, Baohua Lai, Xiaoguang Hu, et al. 2022. Pp-ocrv3: More attempts for the improvement of ultra lightweight ocr system. *arXiv preprint arXiv:2206.03001*.

Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. 2019. Tablebank: A benchmark dataset for table detection and recognition. *arXiv preprint arXiv:1903.01949*.

Minghao Li, Yiheng Xu, Leyang Cui, Shaohan Huang, Furu Wei, and Zhoujun Li. 2020. Docbank: A benchmark dataset for document layout analysis. *arXiv preprint arXiv:2006.01038*.

Yiren Li, Zheng Huang, Junchi Yan, Yi Zhou, Fan Ye, and Xianhui Liu. 2021. Gfte: graph-based financial table extraction. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part II*, pages 644–658. Springer.

Nikolaos Livathinos, Cesar Berrospi, Maksym Lysak, Viktor Kuropiatnyk, Ahmed Nassar, Andre Carvalho, Michele Dolfi, Christoph Auer, Kasper Dinkla, and Peter Staar. 2021. Robust pdf document conversion using recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 15137–15145.

Sabri A Mahmoud, Irfan Ahmad, Wasfi G Al-Khatib, Mohammad Alshayeb, Mohammad Tanvir Parvez, Volker Märgner, and Gernot A Fink. 2014. Khatt: An open arabic offline handwritten text database. *Pattern Recognition*, 47(3):1096–1112.

Sabri A Mahmoud, Hamzah Luqman, Baligh M Al-Helali, Galal BinMakhashen, and Mohammad Tanvir Parvez. 2018. Online-khatt: an open-vocabulary database for arabic online-text processing. *The Open Cybernetics & Systemics Journal*, 12(1).

Hamdy Mubarak, Hend Al-Khalifa, and AbdulMohsen Al-Thubaity. 2022. Overview of osact5 shared task on arabic offensive language and hate speech detection. In *Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 162–166.

Ahmed Nassar, Nikolaos Livathinos, Maksym Lysak, and Peter Staar. 2022. Tableformer: Table structure understanding with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4614–4623.

Shubham Singh Paliwal, D Vishwanath, Rohit Rahul, Monika Sharma, and Lovekesh Vig. 2019. Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 128–133. IEEE.

Werner Pantke, Martin Dennhardt, Daniel Fecker, Volker Märgner, and Tim Fingscheidt. 2014. An historical handwritten arabic dataset for segmentation-free word spotting-hadara80p. In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 15–20. IEEE.

Vik Paruchuri. 2024a. Marker: Convert pdf to markdown and other formats. https://github.com/VikParuchuri/marker.

Vik Paruchuri. 2024b. Surya: Accurate line-by-line text detection and recognition in complex documents. https://github.com/VikParuchuri/surya.

Mario Pechwitz, S Snoussi Maddouri, Volker Märgner, Noureddine Ellouze, Hamid Amiri, et al. 2002. Ifn/enit-database of handwritten arabic words. In *Proc. of CIFED*, volume 2, pages 127–136. Citeseer.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S Nassar, and Peter W J Staar. 2022. Doclaynet: A large human-annotated dataset for document-layout analysis. *arXiv preprint arXiv:2206.01062*.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395.

Devashish Prasad, Ayan Gadpal, Kshitij Kapadni, Manish Visave, and Kavita Sultanpure. 2020. Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 572–573.

Rana SM Saad, Randa I Elanwar, NS Abdel Kader, Samia Mashali, and Margrit Betke. 2016. Bce-arabic-v1 dataset: Towards interpreting arabic document images for people with visual impairments. In *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, pages 1–8.

M. Saeed, A. Chan, A. Mijar, and J. Moukarzel. 2024. Muharaf: Manuscripts of handwritten arabic dataset for cursive text recognition. *arXiv preprint arXiv:2406.09630*.

Sebastian Schreiber, Stefan Agne, Ivo Wolf, Andreas Dengel, and Sheraz Ahmed. 2017. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 1162–1167. IEEE.

Zejiang Shen, Ruochen Zhang, Melissa Dell, Benjamin Charles Germain Lee, Jacob Carlson, and Weining Li. 2021. Layoutparser: A unified toolkit for deep learning based document image analysis. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I 16*, pages 131–146. Springer.

Fouad Slimane, Rolf Ingold, Slim Kanoun, Adel M Alimi, and Jean Hennebert. 2009. A new arabic printed text image database and evaluation protocols. In *2009 10th international conference on document analysis and recognition*, pages 946–950. IEEE.

Ray Smith. 2007. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE.

Peter WJ Staar, Michele Dolfi, Christoph Auer, and Costas Bekas. 2018. Corpus conversion service: A machine learning platform to ingest documents at scale. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 774–782.

Jingqun Tang, Qi Liu, Yongjie Ye, Jinghui Lu, Shu Wei, Chunhui Lin, Wanqing Li, Mohamad Fitri Faiz Bin Mahmood, Hao Feng, Zhen Zhao, Yanjie Wang, Yuliang Liu, Hao Liu, Xiang Bai, and Can Huang. 2024. Mtvqa: Benchmarking multilingual text-centric visual question answering. *Preprint*, arXiv:2405.11985.

Qwen Team. 2025. Qwen2.5-vl.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, et al. 2024a. Mineru: An open-source solution for precise document content extraction. *arXiv preprint arXiv:2409.18839*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024b. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Maurice Weber, Carlo Siebenschuh, Rory Butler, Anton Alexandrov, Valdemar Thanner, Georgios Tsolakis, Haris Jabbar, Ian Foster, Bo Li, Rick Stevens, et al. 2023. Wordscape: a pipeline to extract multilingual, visually rich documents with layout annotations from web crawl data. *Advances in Neural Information Processing Systems*, 36:26048–26068.

Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, et al. 2024. General ocr theory: Towards ocr-2.0 via a unified end-to-end model. *arXiv preprint arXiv:2409.01704*.

Yue Wu and Prem Natarajan. 2017. Self-organized text detection with minimal post-processing via border learning. In *International Conference on Computer Vision*.

Renqiu Xia, Bo Zhang, Haoyang Peng, Hancheng Ye, Xiangchao Yan, Peng Ye, Botian Shi, Yu Qiao, and Junchi Yan. 2023. Structchart: Perception, structuring, reasoning for visual chart understanding. *arXiv preprint arXiv:2309.11268*.

Renqiu Xia, Bo Zhang, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Min Dou, Botian Shi, Junchi Yan, et al. 2024. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *arXiv preprint arXiv:2402.12185*.

Taha Zerrouki and Amar Balla. 2017. Tashkeela: Novel corpus of arabic vocalized texts, data for auto-diacritization systems. *Data in brief*, 11:147.

Y Zhao, W Lv, S Xu, J Wei, G Wang, Q Dang, Y Liu, and J Chen. 2023. Detrs beat yolos on real-time object detection. arxiv e-prints. *arXiv preprint arXiv:2304.08069*.

Z. Zhao, H. Kang, B. Wang, and C. He. 2024. Doclayout-yolo: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception. *arXiv preprint arXiv:2410.12628*.

Xinyi Zheng, Douglas Burdick, Lucian Popa, Xu Zhong, and Nancy Xin Ru Wang. 2021. 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.

X Zhong, E ShafieiBavani, and A Jimeno-Yepes. 2019a. Image-based table recognition: data, model, and evaluation. corr abs/1911.10683. *arXiv preprint arXiv:1911.10683*.

Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. 2020. Image-based table recognition: data, model, and evaluation. In *European conference on computer vision*, pages 564–580. Springer.

Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. 2019b. Publaynet: largest dataset ever for document layout analysis. In *2019 International conference on document analysis and recognition (ICDAR)*, pages 1015–1022. IEEE.

A Source of the Existing Dataset Collection

Our benchmark integrates diverse data sources to ensure comprehensive coverage of Arabic document types. As detailed in Table 2, the dataset combines manually curated samples, synthetic data generated through our LLM-assisted pipeline (Figure 4), and existing publicly available datasets. Key sources include:

- Handwritten Text: KHATT (paragraph and line-level annotations), ADAB, Muharaf, and OnlineKhatt.
- Historical Documents: HistoryAr and HistoricalBooks.
- Scene Text: EvAREST for real-world context diversity.
- Layout Analysis: BCE-Arabic-v1 and DocLayNet.
- Synthetic Content: 576 chart samples (16 types) and 422 diagram samples generated via our five-phase pipeline (Section 3.2).

The dataset emphasizes domain diversity, covering academic, medical, legal, financial, and technical documents. All samples underwent rigorous validation by native Arabic speakers to ensure linguistic and structural accuracy.

B Detailed Performance Comparison

Table 9 provides granular performance metrics for VLMs and OCR frameworks across 12 Arabic text recognition datasets. Gemini-2.0-Flash demonstrates exceptional robustness on synthetic datasets (CER: 0.01 on PATS), while AIN-7B excels in historical manuscript recognition (CER: 0.26 on HistoryAr). Traditional OCR systems like Tesseract show limitations in handwritten text (CER: 1.26 on HistoryAr), highlighting the need for script-specific optimizations.

C Data Analysis

Our data generation pipeline (Figure 4) enabled the creation of 1,502 synthetic samples (576 charts, 422 diagrams, 456 tables). The pipeline's human validation phase rejected 18% of initial outputs due to RTL formatting errors or semantic inconsistencies. As shown in Figure 5 and 6, domain-specific prompts ensured adherence to Arabic linguistic conventions during LLM-assisted generation. The final dataset exhibits balanced representation across:

- Font Styles: 21 Arabic calligraphic styles
- Document Types: 36 sub-domains including financial reports and technical manuals
- Structural Complexity: 43% of tables contain merged cells; 29% of charts use dual-axis configurations

D Tasks Models and Metrics

Table 10 maps evaluation tasks to corresponding models and metrics. The framework evaluates nine core capabilities:

- Structural Understanding: Layout detection (mAP), line detection (IoU)
- Content Extraction: Text recognition (CER), table parsing (TEDS)
- Semantic Reasoning: VQA accuracy, chart-to-dataframe conversion (SCRM)
- Specialized metrics like MARS (α=0.5) address the dual requirements of text fidelity and structural preservation in PDF-to-Markdown conversion.

E SCRM and CODM

The Structuring Chart-oriented Representation Metric (SCRM) evaluates chart understanding through three components:

$$\text{SCRM} = 0.4J_{\text{type}} + 0.3J_{\text{topic}} + 0.3J_{\text{data}} \quad (4)$$

where J_{type}, and J_{topic} are chrF scores, and J_{data} measures JSON structural similarity.

The Code-Oriented Diagram Metric (CODM) extends SCRM for flowcharts and technical diagrams:

$$\text{CODM} = 0.5J_{\text{topology}} + 0.5J_{\text{semantics}} \quad (5)$$

assessing both node-edge relationships and semantic labels. As shown in Figure 5 and 6, domain-specific prompts guided model responses for metric calculation. For instance, sequence diagrams required strict adherence to Arabic UML notation standards during evaluation.

Domain	**Sub-Domain**	**Dataset Source**	**Original**	**Selected**	**Total**
PDF to Markdown	General	Manual	33	33	33
Layout Detection	Docs	BCE-Arabic-v1 (Saad et al., 2016)	1.9k	1,700	2,100
		DocLayNet (Pfitzmann et al., 2022)	80k	400	
Line Detection	Docs	Manual	375	378	378
Line Recognition	Docs	Manual	375	378	378
Table Recognition	Financial	Pixmo (Deitke et al., 2024)	490	456	456
Image to Text	Synthetic	PATS (El-Muhtaseb, 2010)	21.6k	500	3,760
		SythenAR	39.1k	500	
	Historical	HistoryAr (Pantke et al., 2014)	1.5k	200	
		HistoricalBooks	40	10	
	Hand. Paragraph	Khatt (Mahmoud et al., 2014)	2.72k	200	
	Hand. Word	ADAB (Boubaker et al., 2021)	15k	200	
		Muharaf (Saeed et al., 2024)	24.5k	200	
	Hand. Line	OnlineKhatt (Mahmoud et al., 2018)	8.5k	200	
		Khatt (Mahmoud et al., 2014)	13.4k	200	
	PPT	ISI-PPT (Wu and Natarajan, 2017)	86.5k	500	
	Blogs	ArabicOCR	20.3k	50	
		Hindawi (Elfilali, 2023)	79k	200	
	Scene	EvAREST (Hassan et al., 2021)	5.59k	800	
Charts to DataFrame	Bar	Synthetic	100	61	576
	Line	Synthetic	100	43	
	Pie	Synthetic	100	56	
	Box	Synthetic	100	31	
	Violin	Synthetic	100	36	
	Area	Synthetic	50	29	
	SunBurst	Synthetic	30	15	
	Dot	Synthetic	30	15	
	Dual Axis	Synthetic	20	26	
	Density Curve	Synthetic	10	5	
	Bubble	Synthetic	20	13	
	Grouped Bar	Synthetic	50	60	
	Stacked Bar	Synthetic	50	82	
	Histogram	Synthetic	100	70	
	HeatMap	Synthetic	10	11	
	Scatter	Synthetic	100	23	
Diagram to Json	Sequence	Synthetic	50	46	226
	Funnel	Synthetic	20	52	
	Class	Synthetic	20	30	
	Network	Synthetic	20	18	
	Venn	Synthetic	20	7	
	FlowChart	Synthetic	100	112	
	TreeMap	Synthetic	100	157	
VQA	Diagrams	Manual	102	102	902
	Charts	Manual	105	100	
	News Letter	PATD (Bouressace and Csirik, 2019)	2.42k	200	
	Scene	MTVQA	818	500	
Total Dataset Size			–		8,809

Table 8: Dataset Distribution Across Different Domains, sub-domains and Data Source

Dataset	Size	GPT-4o		GPT-4o-mini		Gemini-2.0-Flash		Qwen2-VL	
		CER	WER	CER	WER	CER	WER	CER	WER
PATS	500	0.23	0.30	0.53	0.71	0.01	0.02	1.02	1.02
SythenAR	500	0.09	0.20	0.14	0.32	0.07	0.17	0.59	1.13
HistoryAr	200	0.51	0.82	0.67	0.96	0.28	0.64	3.46	2.86
HistoricalBooks	10	0.41	0.76	0.59	0.88	0.05	0.22	1.90	2.16
Khatt	200	0.45	0.74	0.64	0.91	0.19	0.45	1.12	5.04
Adab	200	0.30	0.73	0.35	0.83	0.19	0.56	0.63	1.08
Muharaf	200	0.56	0.90	0.63	0.94	0.33	0.69	3.57	2.87
OnlineKhatt	200	0.29	0.63	0.41	0.76	0.17	0.44	1.30	2.01
ISI-PPT	500	0.08	0.18	0.15	0.31	0.06	0.15	1.03	1.06
ArabicOCR	50	0.06	0.26	0.16	0.46	0.00	0.02	1.25	1.50
Hindawi	200	0.34	0.56	0.48	0.71	0.01	0.04	1.82	2.05
EvArest	800	0.20	0.38	0.25	0.51	0.18	0.36	0.41	0.95
	3,760	0.31	0.55	0.43	0.71	0.13	0.32	1.48	1.20

Dataset	Size	Qwen2.5-VL		AIN		Tesseract		Surya	
		CER	WER	CER	WER	CER	WER	CER	WER
PATS	500	0.26	0.36	0.00	0.00	0.14	0.28	4.66	4.67
SythenAR	500	0.21	0.40	0.04	0.16	0.31	0.72	4.82	7.90
HistoryAr	200	0.47	0.83	0.26	0.54	0.72	1.26	10.32	12.78
HistoricalBooks	10	0.33	0.72	0.84	0.88	0.74	0.99	6.81	6.30
Khatt	200	0.07	0.22	0.61	1.12	0.67	1.06	4.25	3.77
Adab	200	0.00	0.01	1.00	1.00	1.00	1.14	7.28	8.71
Muharaf	200	0.61	0.96	0.38	0.54	0.77	1.22	6.19	7.48
OnlineKhatt	200	0.36	0.70	0.03	0.12	0.59	1.20	6.71	6.95
ISI-PPT	500	0.36	0.54	0.52	0.53	0.31	0.64	4.25	3.77
ArabicOCR	50	1.00	1.00	0.01	0.01	0.01	0.01	2.75	3.58
Hindawi	200	1.00	1.00	0.11	0.15	0.31	0.72	0.15	0.20
EvArest	800	0.19	0.36	0.30	0.32	0.85	1.02	5.91	3.86
	3,760	0.28	0.54	0.20	0.58	0.89	0.79	4.95	5.61

Table 9: Performance comparison of Large Vision-Language Models on KITAB-Bench (lower is better).

Task	**Metrics**	**Open LLMs**	**Closed LLMs**	**OCR Systems**
Document Understanding Tasks				
PDF to Markdown	chrF + TEDS	–	–	Docling Marker MinerU PDF-Extract-Kit
Layout Detection	mAP@0.5 mAP@0.5:0.95 Precision Recall F1	–	–	Surya Yolo-doclaynet (MinerU) Detr (docling)
Line Detection	mAP@0.5 mAP@0.5:0.95	–	–	Surya Tesseract EasyOCR
Line Recognition	WER, CER	–	–	Surya Tesseract EasyOCR
Table Understanding Tasks				
Tables Recognition (HTML)	TEDS (Zhong et al., 2019a)	Qwen2-VL Qwen2.5-VL AIN PaliGemma	GPT-4o GPT-4o-mini Gemini-2.0-Flash	Docling[EasyOCR] Docling[Tesseract] Marker Img2Table[EasyOCR] Img2Table[Tesseract]
Tables Recognition (CSV)	Jaccard Index	Qwen2-VL Qwen2.5-VL AIN PaliGemma	GPT-4o GPT-4o-mini Gemini-2.0-Flash	Docling[EasyOCR] Docling[Tesseract] Marker Img2Table[EasyOCR] Img2Table[Tesseract]
Visual Understanding Tasks				
Image to Text	CER, WER chrF, BLEU METEOR	Qwen2-VL Qwen2.5-VL AIN-7B PaliGemma	GPT-4o GPT-4o-mini Gemini-2.0-Flash	Docling[EasyOCR] Docling[Tesseract] Marker Img2Table[EasyOCR] Img2Table[Tesseract]
Charts to DataFrame	SCRM (Xia et al., 2024, 2023)	Qwen2-VL Qwen2.5-VL AIN PaliGemma	GPT-4o GPT-4o-mini Gemini-2.0-Flash	–
Diagram to Json	SCRM	Qwen2-VL Qwen2.5-VL AIN-7B PaliGemma	GPT-4o GPT-4o-mini Gemini-2.0-Flash	–
VQA	Accuracy + Word Match Score	Qwen2-VL Qwen2.5-VL AIN-7b PaliGemma	GPT-4o GPT-4o-mini Gemini-2.0-Flash	–

Table 10: Comprehensive evaluation metrics and models for document understanding tasks. The table is organized into three main categories: document understanding, table understanding, and visual understanding tasks. Each task is evaluated using specific metrics and implemented across various models and OCR systems.

Charts: Type Prompt

"""You are an expert in detecting chart types. Below are examples of the expected output format:

Example 1:
bar chart

Example 2:
scatter chart

Example 3:
histogram

Your task is to determine the type of chart shown in the given image.

Instructions:
- **Respond with only the chart type** (e.g., 'bar chart', 'scatter chart').
- **Do not include any additional text, explanations, or descriptions.**
- **Ensure the output matches the format in the examples exactly.**

Provide only the chart type in **single quotes** as shown in the examples above.

What type of chart is shown in the image? Don't output any extra text"""

Charts: Topic Prompt

"""أنت خبير في تحليل وتقييم المخططات البيانية. فيما يلي أمثلة توضح تنسيق الإجابة المتوقع:

مثال 1:
توزيع الكتب الأكثر مبيعاً حسب النوع الأدبي

مثال 2:
آراء العملاء حول الموضوعات المثيرة للجدل في الكتب

التعليمات:
- **حدد موضوع أو محتوى المخطط البياني فقط.**
- **اكتب الإجابة باللغة العربية فقط.**
- **اتبع التنسيق المحدد دون إضافة أي شرح أو تعليق إضافي.**

ما هو موضوع أو محتوى المخطط البياني؟"""

Charts: Data Prompt

"""You are an expert in chart data extraction. You are given a chart image and you should provide the chart data in CSV format.
Here are some examples.
Example 1:
```csv
النوع الأدبي,المبيعات (بالآلاف)
روايات,٣٥٠
خيال علمي,١٢٠
فانتازيا,١٨٠
حياتي,٩٠
تاريخ,٧٠
علم نفس,١١٠
مذكرات,٨٥
تكنولوجيا,٦٥
فنون,٤٥
أطفال,٢٠٠
```

Example 2:
```csv
موضوع,نسبة العملاء الإيجابية,نسبة العملاء السلبية
السياسة في الأدب,٦٠,٤٠
الدين والفكر,٦٥,٣٥
العلاقات غير التقليدية,٤٥,٥٥
العنف في القصص,٧٠,٣٠
الحريات الفردية,٥٠,٥٠
النقد الاجتماعي,٤٠,٦٠
التكنولوجيا والمستقبل,٣٥,٦٥
```
Not give me the results as in the previous CSV format."""

PDF to Markdown Prompt

"""Extract the text from the document in Markdown format, and extract the tables in HTML format.
Do not add style or anything, just the text. Do not ever generate tables in markdown format. Give me the output, nothing else."""

OCR Prompt

"""Extract the text in the image. Give me the final text, nothing else."""

Diagrams: Type Prompt

"""You are an expert in detecting chart types. Below are examples of the expected output format:

Example 1:
treemap

Example 2:
flowchart

Example 3:
diagram

Your task is to determine the type of chart shown in the given image.

Instructions:
- **Respond with only the chart type** (e.g., 'flowchart', 'sequence').
- **Do not provide any explanations, descriptions, or additional text.**
- **Ensure the output strictly follows the format shown in the examples.**

What type of chart is shown in the image?"""

Diagrams: Topic Prompt

"""You are an expert in detecting chart types. Below are examples of the expected output format:

Example 1:
treemap

Example 2:
flowchart

Example 3:
diagram

Your task is to determine the type of chart shown in the given image.

Instructions:
- **Respond with only the chart type** (e.g., 'flowchart', 'sequence').
- **Do not provide any explanations, descriptions, or additional text.**
- **Ensure the output strictly follows the format shown in the examples.**

What type of chart is shown in the image?"""

Figure 5: Prompts for Different Task Categories.

Diagrams: Data Prompt

````
"""You are an expert in diagram data extraction. Your task is to
analyze the given diagram and generate structured data in JSON
format that captures nodes (entities) and edges (relationships).

## Output Format Example:
for flowchart:
```json
{
  "nodes": [
    {
      "id": "1",
      "text": "جمع النفايات",
      "description": "جمع النفايات الصلبة من المناطق الحضرية"
    },
    {
      "id": "2",
      "text": "فرز النفايات",
      "description": "فرز النفايات إلى مواد قابلة لإعادة التدوير وغير قابلة"
    },
    {
      "id": "3",
      "text": "نقل النفايات",
      "description": "نقل النفايات غير القابلة للتدوير إلى مرافق التحويل"
    }
  ],
  "edges": [
    {
      "from": "1",
      "to": "2",
      "text": "فرز"
    },
    {
      "from": "2",
      "to": "3",
      "text": "نقل"
    },
    {
      "from": "3",
      "to": "4",
      "text": "معالجة"
    }
  ]
}
treemap:
```json
{
  "تخصيص التمويل الحكومي والمساعدات": {
    "التنمية الاقتصادية": {
      "دعم المشاريع الصغيرة": "تمويل الأسر المتأثرة",
      "تحفيز الاستثمارات المحلية": "إعفاءات ضريبية"
    },
    "البنية التحتية": {
      "تحسين النقل العام": "تحديث الحافلات والقطارات",
      "الصرف الصحي والمياه": "معالجة مياه الصرف"
    }
  }
}
class diagram:
```json
{
  "ورشة_عمل": {
    "خصائص": [
      "اسم_الورشة",
      "تاريخ_الورشة",
      "مدة_الورشة"
    ],
    "علاقات": {
      "تحتوي_على": "جلسة_تدريب",
      "ينظم_من": "منظم",
      "يشارك_في": "مشارك"
    }
  },
  "جلسة_تدريب": {
    "خصائص": [
      "اسم_الجلسة",
      "مدة_الجلسة",
      "محتوى"
    ],
    "علاقات": {
      "يقدم_من": "مدرب",
      "يتبع_إلى": "ورشة_عمل"
    }
  }
}
"""
````

Table: HTML Prompt

````
"""Extract the data from the table below and provide the output
in HTML format. Output only the data as HTML and nothing else.
Here is one example:
```html
<table>
  <thead>
    <tr>
      <th>الفئة</th>
      <th>النسبة المئوية</th>
      <th>التفاصيل</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>الأسهم المحلية</td>
      <td>٣٥٪</td>
      <td>شركة سابك, شركة الاتصالات السعودية, شركة أرامكو</td>
    </tr>
    <tr>
      <td>الأوراق المالية الحكومية</td>
      <td>٢٠٪</td>
      <td>حكومة السعودية, حكومة الإمارات</td>
    </tr>
    <tr>
      <td>السندات الدولية</td>
      <td>١٥٪</td>
      <td>بنك سويسري, بنك جي بي مورغان</td>
    </tr>
    <tr>
      <td>العقارات التجارية</td>
      <td>١٥٪</td>
      <td>دبي, الرياض, المنامة</td>
    </tr>
    <tr>
      <td>الاستثمارات البديلة</td>
      <td>١٠٪</td>
      <td>صناديق الاستثمار الخاصة, صناديق التحوط</td>
    </tr>
    <tr>
      <td>النقد وما يعادله</td>
      <td>٥٪</td>
      <td>بنك الإمارات دبي الوطني, بنك أبوظبي الأول</td>
    </tr>
  </tbody>
</table>
```
Now generate the data for the provided table."""
````

Table: Dataframe Prompt

````
"""Extract the data from the table below and
provide the output in CSV format. Output
only the data as CSV and nothing else. Here
is one example:
```csv
اسم الشركة,الصفقة,مبلغ الصفقة (مليون دولار),تاريخ الاتفاقية,نوع التقنية
أوراكل,الاستحواذ على شركة سيرنر,28,2023-06-15,الحوسبة السحابية والنمذجة الحيوية
أمازون ويب سيرفيسز,شراكة مع شركة موديلينغ بيو,15,2023-04-20,النمذجة الحيوية
مايكروسوفت,شراكة مع شركة بيومادكس,12,2023-03-10,الحوسبة السحابية
جوجل كلاود,شراء شركة بيوكيم سوليوشنز,35,2023-09-01,النمذجة الحيوية
آي بي إم,توسع في شراكتها مع شركة جينوميك سوفتوير,18,2023-05-05,حوسبة بيولوجية

```
Now generate the data for the provided
table."""
````

Figure 6: Prompts for Different Task Categories (Continued).
````