60-second takeaway No single model wins all document types. Qianfan has the lowest aggregate CER (12.8%), but the real insight is per-archetype: GLM dominates diagrams (6.1% CER), Hunyuan wins on low-contrast scans (6.6%), and Qianfan sweeps text, tables, formulas, and worksheets. The right answer is a routing rule, not a single model.

Where this fits

- **For founders:** if you are building an OCR pipeline, do not pick one model - route by document type. This page gives you the routing table with the data behind it, so you can skip the bakeoff and ship.
- **For engineers:** the per-archetype CER tables below are the decision input for a page-level router. Use them to set your routing thresholds, pick fallback models, and avoid the models that hallucinate on your document mix.

Why scanned PDFs are hard

Born-digital PDFs have extractable text layers. You can copy-paste from them, search them, and feed them straight into downstream pipelines.

Scanned PDFs are images. Every page is a raster - OCR must reconstruct text from pixels. That reconstruction fails in predictable ways:

- **Degraded scans** - faded ink, uneven lighting, skewed pages, coffee stains. The model sees noise where you see text.
- **Complex layouts** - multi-column pages, nested tables, sidebars, footnotes. The reading-order problem is as hard as the character-recognition problem.
- **Formulas** - mathematical notation requires spatial reasoning that most OCR models were not trained for. A subscript in the wrong place changes the meaning entirely.
- **Diagrams with embedded text** - flowcharts, circuit diagrams, annotated figures. The model must separate diagram elements from readable text.

- **Handwritten content** - annotations, margin notes, filled-in worksheets. Most models trained on printed text struggle here.

The failure mode is not "OCR returns nothing." The failure mode is OCR returns plausible-looking text that is wrong - and you do not notice until a downstream consumer breaks.

Test methodology

We used the **lightonocr-slice-v1** corpus: 50 pages drawn from real scanned PDFs, classified into 7 archetypes by visual structure.

Corpus breakdown

Archetype	Pages	What it tests
text_first_notes	10	Clean printed text, minimal layout complexity
diagram_question	10	Inline diagrams with embedded text labels
table_heavy	8	Multi-row, multi-column tabular data
formula_heavy	8	Mathematical notation (LaTeX-level complexity)
worksheet_options	8	Multiple-choice layouts, numbered items
blank_or_near_blank	3	Pages with little or no content (false positive test)
low_contrast_or_faint_scan	3	Degraded, faded, or low-contrast scans

Models tested

Five open OCR models, each run on every page:

1. **Qianfan** (Baidu)
2. **GLM** (Zhipu AI)
3. **Hunyuan** (Tencent)
4. **FireRed** (FireRed AI)
5. **DeepSeek** (DeepSeek)

Evaluation method

We computed **CER** (Character Error Rate) and **WER** (Word Error Rate) using cross-model consensus as the reference. Where no human ground truth exists, the highest-consensus model output serves as the reference string. This is not a perfect proxy - but it is a practical one that scales to hundreds of pages without manual transcription.

For interactive side-by-side comparison and per-page voting, see the internal /ocr-review tool.

Results: aggregate comparison

Model	CER (%)	WER (%)
Qianfan	**12.8**	**13.18**
GLM	33.84	27.59
Hunyuan	35.5	29.3
FireRed	39.01	23.88
DeepSeek	39.34	33.39

Qianfan leads by a wide margin on aggregate CER. But aggregate scores hide archetype-specific performance. GLM, for example, is 3.6x better than Qianfan on diagram pages - a fact invisible in the aggregate table.

The per-archetype breakdown below is where the routing decisions come from.

Results by document type

Per-archetype CER (%) by model

Archetype	Pages	FireRed	GLM	Hunyuan	DeepSeek	Qianfan
text_first_notes	10	10.0	20.7	8.2	8.3	**5.9**
diagram_question	10	39.9	**6.1**	65.9	30.2	22.0
formula_heavy	8	78.7	108.6	42.5	76.6	**20.7**
table_heavy	8	39.7	35.6	63.6	43.8	**15.7**
worksheet_options	8	12.2	15.7	16.2	46.5	**7.1**
low_contrast_or_faint_scan	3	16.3	14.2	6.6	69.1	**0.0**
blank_or_near_blank	2	158.8	N/A	**0.0**	**0.0**	**0.0**

Bold marks the best model per archetype.

Text-first notes

All models perform reasonably on clean printed text. Qianfan is best at 5.9% CER. Hunyuan (8.2%) and DeepSeek (8.3%) are close behind. GLM lags at 20.7% - acceptable for many use cases, but not best-in-class for this archetype.

Takeaway: for straightforward text pages, any model works. Qianfan and Hunyuan are the safest picks.

Diagram questions

GLM dominates at 6.1% CER - nearly 4x better than Qianfan (22.0%) and over 10x better than Hunyuan (65.9%). Hunyuan struggles badly with inline diagrams, likely confusing diagram elements with text.

Takeaway: use GLM for any page with inline diagrams, flowcharts, or annotated figures.

Formula-heavy

Qianfan wins at 20.7% CER. Hunyuan is a distant second at 42.5%. GLM is the worst at 108.6% - a CER above 100% means the model hallucinated more characters than exist in the reference. GLM actively fabricates content when it encounters formulas.

Takeaway: use Qianfan for mathematical content. Avoid GLM entirely on formula pages.

Table-heavy

Qianfan again leads at 15.7% CER. GLM (35.6%) and FireRed (39.7%) are in the middle. Hunyuan is worst at 63.6% - it struggles with multi-column alignment and cell boundaries.

Takeaway: Qianfan for tables. GLM is an acceptable runner-up.

Worksheet/options

Qianfan best at 7.1%. FireRed (12.2%) and GLM (15.7%) are reasonable. DeepSeek is worst at 46.5% - it misreads option labels and numbering.

Takeaway: Qianfan or FireRed for multiple-choice and worksheet layouts.

Low-contrast / faint scans

Qianfan achieves 0.0% CER on the low-contrast pages in this corpus. Hunyuan is good at 6.6%. DeepSeek is terrible at 69.1% - it fails to extract legible text from degraded scans.

Takeaway: Qianfan handles degraded scans best. Hunyuan is the fallback.

Blank / near-blank pages

Hunyuan, DeepSeek, and Qianfan all correctly return empty or near-empty output (0.0% CER). FireRed hallucinates text on blank pages - 158.8% CER means it generated far more text than exists on the page.

Takeaway: if your pipeline processes blank pages (common in batch-scanned documents), avoid FireRed. Use Hunyuan or DeepSeek as blank-page detectors.

The routing decision tree

Based on the per-archetype data above, here is the routing table we use:

Document type	Best model	Runner-up	Avoid
Text-first notes	Qianfan	Hunyuan	-
Diagram questions	GLM	Qianfan	Hunyuan
Formula-heavy	Qianfan	Hunyuan	GLM
Table-heavy	Qianfan	GLM	Hunyuan
Worksheets	Qianfan	FireRed	DeepSeek
Low-contrast scans	Qianfan	Hunyuan	DeepSeek
Blank pages	Hunyuan or DeepSeek	Qianfan	FireRed
Mixed (unknown type)	Route by archetype	Qianfan as fallback	-

The "Avoid" column is not theoretical. GLM on formulas hallucinates. FireRed on blank pages hallucinates. DeepSeek on degraded scans returns garbage. These are not edge cases - they are systematic failures that a routing rule prevents.

Processing speed comparison

Model	Latency (s/page)	Relative speed
GLM	**0.9**	1x (baseline)
FireRed	3.4	3.8x slower
Hunyuan	6.6	7.3x slower
DeepSeek	14.8	16.4x slower

GLM at 0.9 seconds per page is 16x faster than DeepSeek at 14.8 seconds. Qianfan latency data was not available for this benchmark run.

The speed/accuracy tradeoff varies by archetype. GLM is fast *and* best on diagrams - but worst on formulas. If your document mix is diagram-heavy, GLM gives you both speed and accuracy. If your mix is formula-heavy, the fastest accurate option is Qianfan.

For batch processing pipelines where latency matters less than accuracy, optimise for CER. For real-time or interactive use cases, GLM's speed advantage is significant.

How to build a routing pipeline

The routing table above assumes you know the document type before calling OCR. In practice, you need a classifier upstream.

Option 1: histogram-based classifier. Compute image-level features - text density, line spacing, presence of large non-text regions - and classify into archetypes with simple heuristics. Fast, no GPU required, works for coarse routing.

Option 2: lightweight vision model. Run a small vision model (or the first few layers of a larger one) to classify the page archetype. More accurate than histograms, but adds latency and cost.

Option 3: two-pass OCR. Run a fast model (GLM at 0.9s/page) first, then decide based on the output whether to re-run with a more accurate model. For example: if GLM output contains LaTeX-like sequences, re-run with Qianfan.

The instavar.com OCR router uses a variant of option 1 combined with archetype-specific confidence thresholds. For the implementation details and how this connects to the broader pipeline, see the hub page.

FAQ

Which single model should I use if I can only run one?

Qianfan. It has the lowest aggregate CER (12.8%) and wins 5 out of 7 archetypes. Its only weakness is diagrams, where GLM is 3.6x better. If your document mix includes few diagrams, Qianfan is the safe default.

Is Tesseract still relevant?

For clean printed text with simple layouts, Tesseract is still functional and free. For scanned documents with complex layouts, degraded quality, formulas, or tables, Tesseract falls behind the models tested here by a wide margin. If you are building a new pipeline in 2026, start with one of the models above.

What about commercial OCR APIs like Mistral OCR 3 or Reducto?

They are viable alternatives if you do not want to self-host. We did not include them in this benchmark because the focus was on open models you can run on your own infrastructure. A commercial API comparison is a separate evaluation with different constraints (cost per page, data residency, rate limits).

How do I evaluate OCR quality on my own documents?

Compute CER and WER against a reference. If you have human-transcribed ground truth, use that. If you do not, use cross-model consensus - run multiple models on the same page and use the highest-agreement output as the reference. Supplement with qualitative spot-checking on edge cases (formulas, tables, degraded scans). The `/ocr-review` tool we built does exactly this.

Can I combine multiple OCR models?

Yes - that is the entire point of this article. Route by document archetype to the model that performs best on that type. The routing table above is the decision input. The engineering cost is a page classifier plus model dispatch logic; the accuracy gain is substantial.

Why is CER above 100% in some cells?

CER measures the edit distance between the OCR output and the reference, normalised by the reference length. A CER above 100% means the model produced more erroneous characters than the reference contains - typically because it hallucinated text that does not exist on the page. GLM at 108.6% on formulas and FireRed at 158.8% on blank pages are both hallucination failures.

Sources

- **OCR model hub:** Which OCR Model Fits Which Workflow in 2026 - the routing-level decision page covering all models we track.
- **Open Document AI Leaderboard:** OCR SOTA Feb 2026 - broader market map with model positioning and workflow-fit analysis.
- **Benchmark methodology:** How We Benchmark OCR Models on Scan-Heavy PDFs - corpus design, scoring method, and the limits of cross-model consensus.