

## 60-second takeaway

Neither Hunyuan OCR nor FireRed OCR is universally better. Across 50 scanned pages in 7 document archetypes, Hunyuan wins 4 out of 7 (text notes, formulas, low-contrast scans, blank pages) while FireRed wins 3 out of 7 (diagrams, tables, worksheets). The split is architectural: Hunyuan's coordinate-grounded output helps on degraded and ambiguous pages; FireRed's markdown-first pipeline is cleaner on structured content. FireRed is also nearly 2x faster. The practical answer is to route by page type, not pick one model globally.

## Where this fits

- **For founders:** If you are choosing between Hunyuan and FireRed, the answer is "both, routed by page type." A single-model stack will underperform on at least 3 of the 7 archetypes we tested. The per-archetype table below gives you the routing rules.
- **For engineers:** Use the CER breakdown by archetype to set routing thresholds. FireRed at 3.4 s/page is the speed default; Hunyuan at 6.6 s/page is the accuracy fallback for degraded inputs. Never send blank-detection jobs to FireRed - it hallucinates phantom text.

## Architecture comparison

Hunyuan and FireRed take fundamentally different approaches to document understanding.

**Hunyuan OCR** produces coordinate-grounded output. Every text region comes with bounding boxes, giving downstream systems spatial context for where text lives on the page. This grounding is especially useful for degraded scans where the model needs to distinguish real content from noise, and for blank-page detection where it can confirm the absence of text regions rather than guessing.

**FireRed OCR** produces markdown-first output. The model directly generates structured Markdown - headings, tables, lists - without an intermediate spatial representation. This makes it faster (no bounding-box overhead) and produces cleaner output on pages that already have clear visual structure. The tradeoff is that it lacks spatial grounding, which hurts on ambiguous or degraded inputs.

<b>Dimension</b>	<b>Hunyuan OCR</b>	<b>FireRed OCR</b>
Output format	Coordinate-grounded (bounding boxes)	Markdown-first
Processing speed	6.6 s/page	3.4 s/page
Strength	Degraded scans, spatial reasoning	Structured content, speed
Weakness	Slower, struggles with complex tables	Blank-page hallucination, formula parsing

## Test setup

We evaluated both models on 50 scanned pages drawn from 7 document archetypes: text-first notes, diagram questions, formula-heavy pages, table-heavy pages, worksheet/options pages, low-contrast or faint scans, and blank or near-blank pages.

Evaluation used Character Error Rate (CER) and Word Error Rate (WER) computed via cross-model consensus - the same framework described in our benchmark methodology. Ground truth was established through majority agreement across 5 OCR models, not manual transcription.

Both models processed 49 of the 50 pages (one page was excluded from each run due to processing failures). All runs used the same input images with no preprocessing differences.

For the full methodology, see: [How We Benchmark OCR Models on Scan-Heavy PDFs](#).

## Aggregate results

<b>Metric</b>	<b>Hunyuan OCR</b>	<b>FireRed OCR</b>
CER (overall)	35.5%	39.01%
WER (overall)	29.3%	23.88%
Pages processed	49	49
Avg. latency	6.6 s/page	3.4 s/page

The aggregate numbers are deceptively close. The real story is in the per-archetype breakdown.

# Head-to-head results by archetype

## Per-archetype CER summary

Archetype	Pages	FireRed CER%	Hunyuan CER%	Winner
text_first_notes	10	10.0	8.2	Hunyuan
diagram_question	10	39.9	65.9	FireRed
formula_heavy	8	78.7	42.5	Hunyuan
table_heavy	8	39.7	63.6	FireRed
worksheet_options	8	12.2	16.2	FireRed
low_contrast_or_faint_scan	3	16.3	6.6	Hunyuan
blank_or_near_blank	2	158.8	0.0	Hunyuan

## Text-first notes

### Winner: Hunyuan (8.2% vs 10.0%)

Both models handle clean, text-heavy pages competently. Hunyuan edges ahead by a small margin. At these error rates, the practical difference is minimal for most downstream tasks. If speed matters more than a 1.8 percentage-point CER gap, FireRed is the reasonable choice here.

## Diagram questions

### Winner: FireRed (39.9% vs 65.9%)

FireRed wins by a significant margin on pages with inline diagrams. Hunyuan's coordinate grounding does not translate into better diagram comprehension - the bounding-box approach appears to fragment diagram-adjacent text, inflating errors. FireRed's markdown pipeline handles the mix of text and visual elements more cleanly.

## Formulas

### Winner: Hunyuan (42.5% vs 78.7%)

This is Hunyuan's strongest archetype win. FireRed's markdown pipeline struggles with mathematical notation - LaTeX-style expressions are frequently malformed or truncated in its output. Hunyuan's spatial grounding helps it preserve formula structure, though a 42.5% CER still means significant errors.

Neither model is production-ready for formula-heavy pages without post-processing.

## Tables

**Winner: FireRed (39.7% vs 63.6%)**

Markdown table extraction is FireRed's architectural strength. Its output format naturally maps to table structures, while Hunyuan's coordinate-based approach must reconstruct table relationships from spatial positions - a harder problem. For table-heavy documents, FireRed is the clear default.

## Worksheets

**Winner: FireRed (12.2% vs 16.2%)**

Both models handle structured worksheets with multiple-choice options well, producing relatively low error rates. FireRed has a modest edge. The structured, repetitive layout of worksheets plays to the strengths of markdown-first output.

## Low-contrast scans

**Winner: Hunyuan (6.6% vs 16.3%)**

Hunyuan wins decisively on faded or low-contrast scans. Coordinate grounding helps here: the model can locate text regions even when contrast is poor, while FireRed's pipeline is more likely to miss or misread faint characters. For document processing pipelines that handle aged or photocopied material, this result matters.

## Blank pages

**Winner: Hunyuan (0.0% vs 158.8%)**

This is the most critical finding in the benchmark. Hunyuan correctly identifies blank pages and returns empty output. FireRed generates phantom text on blank pages - hallucinated content that does not exist in the source image. A CER of 158.8% means FireRed produces more characters of hallucinated text than a typical page contains.

This is not a minor edge case. Any document processing pipeline encounters blank pages - separator sheets, blank backs of single-sided prints, intentional

empty pages. A model that hallucinates content on blank pages will inject noise into every downstream system.

## Speed comparison

FireRed processes pages at 3.4 seconds per page - nearly twice as fast as Hunyuan's 6.6 seconds per page. For a 100-page document, that is the difference between roughly 6 minutes and 11 minutes of processing time.

The speed gap is architectural. FireRed's markdown-first pipeline skips bounding-box computation, while Hunyuan must compute spatial coordinates for every detected text region before generating output.

For latency-sensitive applications (real-time document intake, user-facing OCR), FireRed's speed advantage is significant. For batch processing where accuracy matters more than throughput, Hunyuan's slower pace is an acceptable tradeoff.

## When to use which

Document type	Use Hunyuan when...	Use FireRed when...
Text notes	Slight accuracy edge needed	Speed matters more
Diagrams	Avoid	Default choice
Formulas	Default choice	Avoid
Tables	Avoid	Default choice
Worksheets	Acceptable	Default choice
Low-contrast scans	Default choice	Acceptable fallback
Blank detection	Default choice	NEVER (hallucination risk)

The routing logic in practice: classify incoming pages by archetype (using a lightweight vision classifier or heuristic rules), then route to the appropriate model. FireRed is the speed-optimised default for structured content. Hunyuan is the accuracy fallback for degraded inputs, formulas, and blank-page gating.

A blank-detection step before the main OCR pass is worth implementing regardless of which model you use for content extraction. Hunyuan's perfect blank-page detection makes it a strong candidate for this gating role.

# FAQ

## Is one model strictly better than the other?

No. Hunyuan wins 4 of 7 archetypes and FireRed wins 3 of 7. The wins are archetype-dependent and driven by architectural differences, not overall model quality. A routed approach using both models outperforms either one alone.

## Can I use FireRed for everything if I need speed?

You can, with one hard exception: never use FireRed for blank-page detection. Its hallucination on blank pages (158.8% CER) will inject phantom content into your pipeline. For all other archetypes, FireRed is a reasonable single-model choice if you accept higher error rates on formulas and low-contrast scans.

## How does cross-model consensus work as ground truth?

We use majority agreement across 5 OCR models as the reference output instead of manual human transcription. This approach scales better than manual annotation and removes single-model bias. The full methodology is documented in our [benchmark methodology guide](#).

## What about other models like GLM-OCR or DeepSeek?

This post focuses on the Hunyuan vs FireRed comparison. For the broader model landscape including GLM-OCR, DeepSeek-OCR-2, dots.ocr-1.5, and PaddleOCR-VL-1.5, see the [OCR Model Leaderboard](#) and the [workflow-fit guide](#).

## Should I run both models on every page?

Running both models on every page doubles your compute cost and latency. A better approach is to classify pages first and route to the stronger model for that archetype. The exception is high-value documents where you want consensus - running both and comparing output can catch errors that either model alone would miss.

## Sources

- [Which OCR Model Fits Which Workflow in 2026](#) - Hub page with routing recommendations across all tested models

- [OCR Model Leaderboard 2026](#) - Benchmark scores, VRAM requirements, and deployment profiles for the full model field
- [How We Benchmark OCR Models on Scan-Heavy PDFs](#) - Methodology for the cross-model consensus evaluation framework used in this comparison
- [OCR for Scanned PDFs: Practical Guide 2026](#) - End-to-end guide for building a scanned PDF processing pipeline