Simple OCR Benchmark Analysis

Tymo van Rijn

May 2026

Scope

This benchmark compares three OCR approaches on the same 150 manually checked frames:

- EasyOCR
- Tesseract
- Gemini 2.5 Flash

Core benchmark table

Method	Rows with time	Coarse correct	Coarse accuracy
EasyOCR	127 / 150	86 / 150	57.3%
Tesseract	100 / 150	76 / 150	50.7%
Gemini 2.5 Flash	144 / 150	144 / 150	96.0%

Table 1: Coarse = MM:SS-level correctness.

Conditional quality (only when a method gives a time)

Method	Correct / predicted rows	Accuracy on predicted rows
EasyOCR	86 / 127	67.7%
Tesseract	76 / 100	76.0%
Gemini 2.5 Flash	144 / 144	100%

Table 2: This separates coverage from quality.

How the Gemini approach works in this project

1. Crop bottom timestamp strips from each frame.
2. Batch strips into PDF pages.
3. Query Gemini with strict JSON schema output.
4. Parse to project JSONL/CSV with `status` and `time_fa_seconds`.
5. Build temporal training split from timestamp-eligible frames only.

Temporal training rule used now

- A frame is eligible only if it has `status=ok` and numeric `time_fa_seconds`.
- `no_time_found` frames are excluded from temporal model training.
- If JSONL has duplicates for the same $(exam, frame)$, the project currently uses the **first** occurrence as the decision source.

Resulting dataset view

Item	Value
Exams in HyperF_Type split	1877
Fully timestamped exams	1814
Not fully timestamped exams	63

Table 3: For the exact HyperF_Type exam set used in training/evaluation.

Conclusion

Gemini provides the best practical trade-off in this pipeline: high coverage and high coarse correctness. Temporal training should therefore use `config/temporal_hyperftype.json`, which explicitly lists valid frames per exam.