A table can look correct in Markdown while its rows, columns, or numbers are wrong. That makes table OCR a validation problem, not only a transcription problem.

We tested NuExtract3 and FireRed-OCR on the same 40 public-safe, scan-like table crops derived from US Bureau of Labor Statistics data. NuExtract3 preserved all 1,040 truth cells in their expected positions. FireRed remained the better general OCR workflow, but four outputs inserted a title row into the HTML table and shifted the cell alignment.

The result supports a specialist fallback pattern. It does not establish that NuExtract3 is the best table OCR model in general.

The short version

- Both models completed all 40 table crops with no failed or empty outputs.
- NuExtract3 strict numeric F1 was 0.963462; FireRed scored 0.957854.
- NuExtract3 matched 1,040/1,040 truth cells and all 480/480 numeric cells after normalization.
- FireRed matched 1,036 cells and 432/480 normalized numeric cells in their expected positions.
- The main FireRed structural failure was placing a title inside the table, shifting later row and column positions.
- A validator routed 23 risky crops to NuExtract3 and kept FireRed for 17 crops.
- The fallback pipeline matched NuExtract3's aggregate score on this sample, but it reused existing outputs and therefore did not measure runtime savings.

What NuExtract3 is

NuExtract3 is a 4B vision-language model for document understanding. Its model card describes structured extraction from text or images, document-to-Markdown conversion, multilingual documents, and HTML table output. The weights are published under Apache-2.0.

That is upstream capability information. Our local result covers only public-safe table crops on one RTX 3090 Ti workflow. It is not evidence for receipts,

contracts, forms, handwriting, or arbitrary financial reports.

Source: NuExtract3 model card.

Why normal OCR metrics are not enough for tables

Character error rate can tell you that text differs from a reference. It cannot tell you whether the value 7.4 stayed in the correct year and series column.

A useful table evaluation needs at least four views:

Check	Failure it catches
Strict numeric-token precision and recall	missing or invented numbers anywhere in the output
Row and column counts	inserted, removed, or merged table structure
Cell-position matching	values shifted into the wrong coordinates
Numeric-cell normalized accuracy	wrong numbers after harmless formatting differences are removed

These scores answer different questions. Keep them separate rather than blending them into one table-accuracy number.

The public-safe test

The sample contained 40 unique table crops rendered from official BLS public data. It covered four data series across 2015 through 2024. Every row had explicit cell truth, numeric-token truth, legal metadata, and public-reporting status.

The crops were intentionally scan-like rather than pristine HTML. Both models received the same images and were scored with the same code.

This was a table-crop diagnostic:

- no Paddle detector was run
- no full-page PDF benchmark was run
- no chemistry documents were included
- no production routing change was made

Results on 40 table crops

Metric	FireRed-OCR	NuExtract3
Successful outputs	40/40	40/40
Failed or empty outputs	0	0
Truth numeric tokens	520	520
Matched numeric tokens	500	501
Missed numeric tokens	20	19
Hallucinated numeric tokens	24	19
Strict numeric precision	0.954198	0.963462
Strict numeric recall	0.961538	0.963462
Strict numeric F1	0.957854	0.963462
Truth cells	1,040	1,040
Matched cells	1,036	1,040
Extra cells	8	0
Row-count matches	36/40	40/40
Column-count matches	36/40	40/40
Cell normalized accuracy	0.900000	1.000000
Numeric-cell normalized accuracy	0.900000	1.000000

The strict numeric F1 difference was small. The cell-position difference was operationally larger.

The failure a visual review can miss

In four FireRed outputs, a title or caption was placed inside the HTML table. The visible table values still appeared afterward, so a quick visual scan could call the output acceptable.

Structurally, every subsequent row was displaced. A downstream CSV import, database loader, or calculation would associate values with the wrong row coordinates.

This is why table evaluation cannot stop at "the numbers are present." It must ask:

- Are the expected rows present?

- Are the expected columns present?
- Did non-table text enter the table?
- Is each number in the expected cell?

NuExtract3 kept all 1,040 cells aligned on this sample. FireRed's four title-row insertions reduced normalized cell and numeric-cell accuracy to 0.9.

Why the strict numeric score tells a different story

The mismatch audit reviewed the 30 highest-severity candidates:

Primary cause	Count
Title or caption numeric error	26
Row or column shift	4

Some title years were misread outside the table body. Those errors correctly reduce strict numeric-token precision or recall, but they do not necessarily corrupt table cells.

That explains the two score views:

- strict numeric F1 checks numeric continuity across the whole output
- numeric-cell accuracy checks values inside the explicit table grid

Neither should replace the other. If titles and captions matter to the application, use both. If only the table body drives calculations, cell-position scoring should control the main acceptance gate.

A validate-then-fallback pipeline

We tested a simple policy that kept FireRed as the general OCR lane and selected NuExtract3 only when validation failed.

The validator requested fallback if any of these occurred:

- row-count mismatch
- column-count mismatch
- numeric-cell normalized accuracy below 0.99
- strict numeric F1 below 0.99

- no parseable table cells
- empty output
- missing cell-level score

On the 40 crops:

- FireRed was accepted for 17 crops
- 23 crops were assigned to NuExtract3
- no crop lacked a usable output
- the selected outputs matched NuExtract3's aggregate scores

Metric	FireRed alone	NuExtract3 alone	Validator-selected output
Strict numeric F1	0.957854	0.963462	0.963462
Cell normalized accuracy	0.900000	1.000000	1.000000
Numeric-cell normalized accuracy	0.900000	1.000000	1.000000
Row-count matches	36/40	40/40	40/40
Column-count matches	36/40	40/40	40/40

This result demonstrates selection logic, not performance savings. The pipeline reused outputs that had already been generated by both models. It did not measure the latency or GPU cost of invoking NuExtract3 only after a FireRed validation failure.

The validator was deliberately conservative

All 23 fallbacks were triggered by strict numeric F1 below 0.99. Only four also had row, column, and numeric-cell failures.

That means the policy can over-route when a title year is wrong but the table body is intact. A production validator should offer at least two modes:

Table-cell-critical mode

Use when calculations or database imports depend on the grid:

- row and column counts
- parseable table cells
- cell-position matching

- numeric-cell accuracy

Broad numeric-continuity mode

Use when titles, dates, captions, footnotes, and table cells all matter:

- all table-cell-critical checks
- strict numeric-token precision and recall across the complete output

The acceptance contract should reflect the downstream risk, not an arbitrary universal threshold.

A practical table OCR checklist

Before accepting extracted table data:

1. Separate page detection, crop selection, transcription, and parsing.
2. Store the raw model output before normalization.
3. Confirm that a table was actually parsed.
4. Compare expected and predicted row and column counts.
5. Keep captions and titles outside the table grid.
6. Normalize harmless formatting differences before cell comparison.
7. Compare numeric values by cell position.
8. Count missing and hallucinated numeric tokens separately.
9. Escalate structural failures to a specialist model or human review.
10. Preserve the validation result with the extracted data.

For unlabelled production documents, some checks need business rules rather than ground truth. Examples include expected column names, numeric ranges, totals, date order, checksum relationships, or reconciliation against a source system.

When to use NuExtract3

NuExtract3 is worth a focused test when:

- table structure is more important than general page transcription
- documents contain forms, invoices, receipts, contracts, or repeated schemas
- HTML or structured extraction is useful downstream

- row shifts and numeric-cell errors are expensive
- a general OCR model already handles ordinary pages

Our result does not justify replacing FireRed for every page. It supports treating NuExtract3 as a table specialist behind an explicit validation gate.

Limitations

- The sample contains 40 BLS-derived scan-like table crops, not complete PDFs.
- It represents four series and ten years, not arbitrary table diversity.
- The comparison does not cover handwriting, merged cells, multi-page tables, rotated pages, or multilingual tables.
- The fallback experiment reused existing outputs and did not measure incremental latency.
- NuExtract3 required a local vLLM 0.17.1 path and tokenizer override in this experiment.
- The result should not be blended with DocLayNet because the DocLayNet slice lacked explicit table text and cell-level numeric truth.

Final verdict

NuExtract3 was the stronger table-cell specialist on this bounded BLS sample. FireRed remained the broader general OCR workflow.

The important result was not the small strict numeric F1 gap. It was the structural difference: NuExtract3 preserved all expected rows, columns, and cell positions, while four FireRed outputs shifted the table by inserting title content.

Use model routing only after defining what "correct table extraction" means for the downstream system. If a wrong number in the right-looking table can trigger a payment, calculation, or report, visual plausibility is not an acceptance test.

For the broader OCR routing map, see Which OCR Model Fits Which Workflow in 2026. For why public OCR leaderboards need local failure testing, read OmniDocBench Is Saturated.