GPU SMITH
COMPUTE ENGINEERING

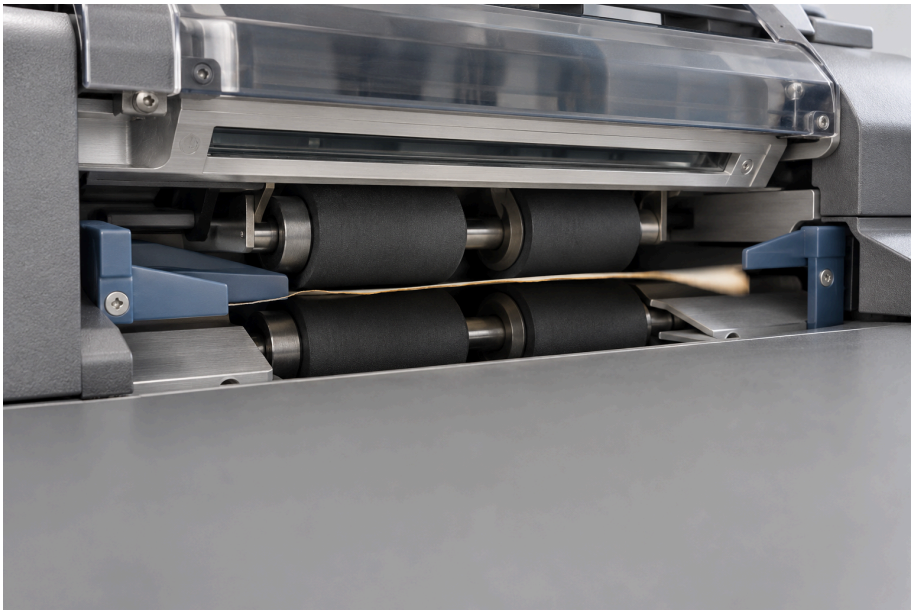
GPU SMITH / RESEARCH & PRACTICAL GUIDANCE

Private OCR Cost per Page: CPU vs GPU vs Textract (2026)

> Plan and build private AI compute with GPU Smith. We connect workload requirements with GPU hardware, networking, deployment and operating decisions so your infrastructure fits the work it must perform.

Published by GPU Smith · 2026-11-02 · private ocr cost per page · self-hosted ocr cost · aws textract pricing per page · gpu vs cpu ocr cost · cost per accepted page · olmocr hardware requirements · paddleocr gpu · on-premise ocr cost comparison · open source ocr vs textract

Original article: https://gpusmith.com/articles/en/private-ocr-cost-cpu-gpu-textract-per-page



In this article

1. Executive Summary
2. Introduction and Background
3. The CPU Path: Digital PDF Extraction and CPU OCR
4. Private GPU OCR and Vision-Language Pipelines
5. Managed Document APIs
6. Feature Comparison
7. Performance and Benchmarks
8. Data Analysis and Evidence
9. Implications and Future Directions
10. Frequently Asked Questions (FAQs)
11. Conclusion

Executive Summary

Private OCR cost per page is not one number. It is a ratio: the full cost of a defined output (plain text, reading-order Markdown, tables, key-value forms or math-preserving transcription) divided by the pages or documents that pass an acceptance test. This report prices that ratio for an illustrative workload (a 1,000,000-page scanned archive due in 14 days, then 100,000 pages per month). Those volumes are scenario assumptions, not customer data, and no throughput, error rate or review time in this report is a measurement.

The published list prices are easy to state. In the worked examples on the AWS pricing page (US West, Oregon), basic Textract text detection costs $0.0015 per page for the first million (aws.amazon.com), and $0.0006 per page beyond that, so two million pages cost $2,100 (aws.amazon.com). Forms plus tables are priced separately at $0.05 plus $0.015 per page in the first tier (aws.amazon.com), and OCR is included in those outputs (aws.amazon.com). Runpod listed an L40S GPU at $1.09 per hour on a page updated October 28, 2026 (www.runpod.io). A continuously billed 730-hour month is $795.70, which equals the basic Textract bill for about 530,467 pages. That is a fixed-bill parity figure only, not a break-even: it ignores throughput, quality, idle time, staff and reserve.

This report does not estimate a CPU cost per page. No CPU instance price or Tesseract or OCRmyPDF pages-per-core-hour figure was obtained, so the CPU path is a buyer-measured input (core-hour price divided by measured accepted pages per core-hour) rather than a priced option, and the comparison below prices the GPU and Textract paths only.

The more useful quantity is the accepted-page throughput a GPU must reach. At $1.09 per hour, compute alone matches the basic Textract rate at roughly 727 accepted pages per replica-hour, and matches the forms-plus-tables rate at roughly 17. Whether a given pipeline reaches either figure is a measurement the buyer has to make on the archive. The olmOCR authors reported under $176 per million pages using February 2025 rental rates, a 12% measured retry rate and an earlier checkpoint (arxiv.org), and the project README repeats an under-$200 headline (github.com). Neither is a quote for this archive.

Review labor can outweigh engine price. At a placeholder $30 loaded hourly rate and 30 seconds per reviewed page, each percentage point of pages sent to review costs $0.0025 per page, which is 1.7 times the basic Textract page price. A multilingual October 2026 study also warns that near-perfect page scores can hide replaced notation (arxiv.org) and that single-page timing may not reflect batched throughput (arxiv.org). The practical path is staged: route born-digital pages through CPU text extraction when they pass the same contract, pilot candidate engines on a human-reviewed gold sample, then rent transient GPU capacity for the backlog before deciding whether steady intake justifies owned hardware. The hypothesis that correction labor can reverse a GPU compute advantage is illustrated here with labeled arithmetic and is untested.

Key numbers (list prices and scenario arithmetic, not measurements):

- Basic Textract text detection: $0.0015 per page for the first million pages; forms plus tables: $0.065 per page.
- Runpod L40S: $1.09 per hour (Secure Cloud) or $0.79 per hour (Community Cloud); $795.70 per continuously billed 730-hour month at the Secure rate.
- Compute-only parity with basic Textract at $1.09 per hour: about 727 accepted pages per replica-hour.
- Each percentage point of pages sent to review: $0.0025 per page at the placeholder labor rate.
- CPU path: not priced here; measure it on the archive.

Introduction and Background

Enterprise document platforms that run private retrieval-augmented generation (RAG) or archive search must turn images of pages into text before anything else happens. That stage precedes chunking, embedding and retrieval. Retrieval hardware is covered separately in the Private RAG Hardware Sizing article, and this report does not repeat it. The question here is narrower: which ingestion path (CPU only, a rented GPU, spare capacity on an existing private GPU server, a newly purchased server, or a managed document API) delivers accepted output at the lowest full cost within the deadline.

GPU Smith is an independent private AI infrastructure advisory. It does not sell OCR software, GPU rentals or a document API, and it appears in this report only as the author's perspective. Nothing here describes a client engagement, a partnership or a benchmark run by the publisher.

Define the output contract first

Page prices for different outputs are not comparable. AWS states that Analyze Document output includes OCR regardless of the feature selected, yet Forms and Tables carry separate per-page charges. The same applies across engines: a plain-text engine, a Markdown-with-reading-order pipeline, a table-structure parser and a key-value form extractor are different products. The olmOCR toolkit, for example, converts PDF, PNG and JPEG documents into Markdown (github.com), while the Textract basic API extracts text and handwriting (aws.amazon.com). Comparing the two per-page prices without checking the output is a category error.

A contract should list, before any benchmarking:

- **Output form:** plain text, reading-order Markdown, table cells with row and column structure, key-value pairs, or LaTeX-style formulas.

- **Critical fields:** the specific fields (identifiers, dates, amounts, names) whose errors make a page or document unacceptable.
- **Languages and scripts:** each with a page count in the archive.
- **Layout:** columns, tables, stamps, marginalia, handwriting, formulas.
- **Privacy boundary:** whether page images may leave the controlled environment at all.
- **Acceptance rule:** page-level, field-level or document-level, and who adjudicates disagreements.

Stratify the archive

Mean accuracy hides concentration. A 1995 evaluation by the Information Science Research Institute found that about 50 to 60% of errors fell on the worst 20% of pages in each sample (tesseract-ocr.github.io). Benchmark designers have reached the same conclusion for modern parsers: OmniDocBench reports results by 9 document types, 6 page-level attributes and 9 bounding-box-level attributes (arxiv.org), and finds a clear accuracy drop on multi-column and complex layouts across all models (arxiv.org). The archive should therefore be split into strata (born-digital versus scanned, language, scan resolution, column count, tables, handwriting, formulas) and each stratum sampled and priced separately.

The CPU Path: Digital PDF Extraction and CPU OCR

Capabilities: a text-layer fast path

Already-digital PDFs contain a text layer, and extracting it needs no OCR. pdfplumber states that it works best on machine-generated rather than scanned PDFs (github.com), and pdfminer.six reads text directly from the PDF source (github.com). pypdfium2 describes itself as a liberal-licensed library for PDF rendering and inspection (github.com). PyMuPDF's OCR call is documented for pages whose text is in raster-image or vector-graphic form (pymupdf.readthedocs.io), which implies that a plain text-layer read is the cheaper route for the others.

The fast path is admissible only if it passes the same contract as OCR output. A text layer can exist and still be wrong (for example, a hidden OCR layer from an earlier scan, or reading order scrambled in columns), so the gold sample must include born-digital pages. OCRmyPDF's skip-text mode illustrates the routing idea: it performs no image processing or OCR on pages that already have text (ocrmypdf.readthedocs.io). The olmOCR toolkit, by contrast, does not make that routing decision. Its pipeline renders every page for the model, and uses pdftotext only as a fallback when a page fails after retries (raw.githubusercontent.com). Sending born-digital pages to a vision-language model by default therefore spends GPU time on pages a CPU could have handled.

Capabilities: CPU OCR

Tesseract is the standard open-source CPU engine. Its documentation states Apache 2.0 terms (tesseract-ocr.github.io) and 100+ languages and 35+ scripts (tesseract-ocr.github.io). Tesseract 4 uses up to four CPU threads per page (tesseract-ocr.github.io), but for bulk work the project recommends one process per CPU core (tesseract-ocr.github.io). GPU support exists only as an OpenCL build that the project does not recommend (tesseract-ocr.github.io). OCRmyPDF wraps this for PDFs, is licensed under MPL-2.0 (github.com) and distributes work across all available CPU cores (github.com).

PaddleOCR's classic pipeline can also run on CPU. Its documentation describes five modules: document orientation, unwarping, text-line orientation, detection and recognition (www.paddleocr.ai). If no device is set, GPU 0 is preferred and the CPU is the fallback (www.paddleocr.ai).

Strengths and limitations

CPU paths scale with cores and need no GPU procurement, driver stack or CUDA compatibility work. Their limits are accuracy on difficult scans and the lack of structured output. In a peer-reviewed 2021 comparison using default settings and no preprocessing, Tesseract ranked last behind Textract and Document AI (link.springer.com) and competed only on noise-free English documents (link.springer.com). That study used single-column test pages, so it did not capture layout parsing (link.springer.com), and the Textract version tested is five years old. It is evidence about noisy scans in 2021, not a ranking for 2026 engines.

Two cost details matter for CPU sizing. First, resolution: Tesseract works best at 300 DPI or above (tesseract-ocr.github.io), and a US Government Publishing Office study found that bitonal scans of older documents gave unacceptable accuracy while grayscale and color did better (www.GovInfo.gov). OCRmyPDF's performance guidance likewise says Tesseract is tuned for roughly 300 dpi (ocrmypdf.readthedocs.io). Second, rendering memory: each concurrent render or OCR job holds its own page image, so RAM, cores and storage bandwidth set a ceiling that can require a separate CPU pool even when inference runs on a GPU.

Licensing is part of the cost. PyMuPDF and MuPDF are available under open-source AGPL or commercial license agreements (pymupdf.readthedocs.io). A buyer that cannot meet AGPL terms for the ingestion service must price the commercial license or choose a permissively licensed extractor.

Private GPU OCR and Vision-Language Pipelines

olmOCR

The Allen Institute for AI's olmOCR toolkit is built on a 7B-parameter vision-language model (VLM), so it requires a GPU (github.com). The README lists a recent NVIDIA GPU (tested on RTX 4090, L40S, A100 and H100) (github.com), at least 12 GB of GPU RAM (github.com) and 30 GB of free disk space. It also needs poppler-utils and fonts to render PDF pages, and a separate vLLM server can host the FP8 olmOCR-2 checkpoint. Rendering is a CPU task: the code shells out to pdftoppm (raw.githubusercontent.com) and bounds render concurrency to avoid overloading the system (raw.githubusercontent.com). The render size is set by a longest-side dimension option.

Licensing for the olmOCR-2 checkpoint is stated as Apache 2.0 on its model card (huggingface.co), which also recommends the FP8 version for practical use (huggingface.co). An earlier card states the model is intended for research and educational use (huggingface.co). The mix of an open license and a stated intended use is the kind of wording a buyer should have reviewed by counsel for the exact checkpoint chosen, rather than inferred from this report.

The cost claim deserves precise reading. The paper says olmOCR can produce high-quality plain text at less than $176 per million pages (arxiv.org). The cost basis is RunPod rental rates (arxiv.org), with the L40S estimated at $0.79 per hour (arxiv.org), a measured 12% retry rate included (arxiv.org), and a test of the launch version of the model (arxiv.org). Runpod's pricing page lists the L40S at $0.79 per hour for Community Cloud

and $1.09 per hour for Secure Cloud, so the paper's rate is a tier that is still listed rather than a superseded price. Substituting the Secure Cloud rate of $1.09 per hour for $0.79 scales that compute figure to roughly $243 per million pages by simple proportion (a Secure-tier sensitivity case), while the Community-tier case stays near the paper's $176. That is arithmetic on someone else's measurement, not a measurement of any other checkpoint, document mix or hardware, and it excludes storage, staff, licensing, reserve and review.

PaddleOCR

PaddleOCR is Apache 2.0 licensed per its repository (github.com). The documented benchmark timings are narrow. The page states that its inference times exclude pre- and post-processing (www.paddleocr.ai), and the test environment was a Tesla T4 with an Intel Xeon Gold 6271C CPU (www.paddleocr.ai) on Ubuntu 20.04, CUDA 11.8, cuDNN 8.9 and TensorRT 8.6.1.6 (www.paddleocr.ai). Per-module milliseconds for detection and recognition under those conditions cannot be converted into end-to-end accepted pages per hour: rendering, orientation handling, retries and serialization are outside them. The documentation also notes that newer PP-OCRv6 accuracy metrics are not directly comparable with earlier versions (www.paddleocr.ai), and that a single PP-OCRv6 model covers 50 languages (www.paddleocr.ai).

The PaddleOCR-VL variant has stricter hardware conditions. Its documentation says vLLM can start on compute-capability 7.x GPUs such as T4 or V100 (www.paddleocr.ai), but that this route is prone to timeouts or out-of-memory failures and is not recommended (www.paddleocr.ai). The Docker route needs NVIDIA drivers supporting CUDA 12.6 or later (www.paddleocr.ai). A buyer planning to reuse older T4-class capacity should therefore test the exact pipeline, not assume the VLM variant will run there.

Surya and Marker

Surya's code and model weights are licensed differently. The repository points to separate terms for broader commercial licensing of the weights (github.com), and the weights are free for research, personal use and startups under $5M funding or revenue (github.com). Marker carries a matching weights term (github.com) and states that it works on GPU, CPU or MPS (github.com). Surya reports 4.5 pages per second on one H100 at 192 DPI (github.com) under specific conditions: a single H100 80 GB, 32 concurrent requests and full-page OCR (github.com). It also advises that lowering render resolution from 192 to 96 DPI improves throughput (github.com). Those are author-reported figures on an expensive accelerator and a particular benchmark set. The lesson for planning is that render DPI is a cost lever and a quality risk at once, and must be fixed in the contract.

Licensing and hardware checks before pricing

- **Code versus weights:** a permissive code license does not settle model-weight terms, as the Surya and Marker examples show.
- **Checkpoint pinning:** record the exact checkpoint, quantization and inference server version, because results and throughput change between releases.
- **Accelerator:** verify the tested GPU. NVIDIA lists the L4 with 72 W low-power operation (www.nvidia.com), and AWS describes G6e instances with up to 8 NVIDIA L40S GPUs and 384 GB of total GPU memory (aws.amazon.com).

- **Host resources:** record CPU cores, RAM and storage bandwidth for rendering, because those can cap the GPU before it saturates.

Managed Document APIs

Amazon Textract

Textract bills per page, with an image counted as a single page (aws.amazon.com). The pricing page's worked examples show the structure: basic text detection at $0.0015 per page for the first million and $0.0006 above it; forms at $0.05 and tables at $0.015 per page, with 5,000 pages costing $325. The examples are for US West (Oregon), and the page text retrieved for this report did not render the regional selector or other regions' tables, so other regions require a check on the live page. A new-customer free tier lasts three months. Tiering is monthly volume based, so a two-million-page bill is not twice the one-million-page bill.

Capability limits affect task equivalence. Printed text, forms and tables are supported in English, German, French, Spanish, Italian and Portuguese (aws.amazon.com), while handwriting, invoices and receipts, identity documents and Queries are English only (aws.amazon.com). Asynchronous PDF and TIFF jobs are limited to 3,000 pages (docs.aws.amazon.com), synchronous operations to 10 MB in memory (docs.aws.amazon.com), and the minimum detectable text height is 15 pixels (docs.aws.amazon.com), equal to 8-point font at 150 DPI (docs.aws.amazon.com). Vertical text is not supported (docs.aws.amazon.com). A multilingual archive with handwriting, vertical text or non-listed scripts needs either a different engine or an explicit exclusion.

Data handling

For a private-document buyer, the data terms matter as much as the price. The FAQ says Textract may store and use document and image inputs to provide and maintain the service (aws.amazon.com), and that some content may be stored in another AWS region (aws.amazon.com). AWS Service Terms (last updated October 30, 2026) say customers may instruct AWS not to use and store AI content processed by an AI service (aws.amazon.com), and the Organizations documentation describes the opt-out policy mechanism (docs.aws.amazon.com). Whether those terms satisfy a given confidentiality regime is a question for the buyer's counsel and security reviewers. A managed API is not a way to keep pages inside the facility.

Other managed options

Prices from other providers are shown for context, not equivalence. Google's Document AI pricing page lists an Enterprise Document OCR processor (cloud.google.com) at $1.50 per 1,000 pages after the first 1,000, falling to $0.60 per 1,000 above 5,000,000, and states that failed requests are not billed (cloud.google.com). Azure Document Intelligence lists Read at $1.50 per 1,000 pages for the first million pages (azure.microsoft.com) and $10 per 1,000 for prebuilt and layout models, with a note that prices are estimates only (azure.microsoft.com). Mistral lists OCR 4.1 at $4 per 1,000 pages (docs.mistral.ai). Each has its own output, region, retention and language terms that must be matched to the contract.

Feature Comparison

Table 1 below compares the candidate path families on the attributes that decide whether they can serve the same contract. It deliberately compares categories and does not rank accuracy.

Path	Output a buyer can expect	License and data boundary	Hardware and capacity notes	Cautions
Born-digital text extraction (pdfminer.six, pdfplumber, pypdfium2, PyMuPDF)	Existing text layer only, no image recognition	MIT-style and permissive options exist; PyMuPDF is AGPL or commercial (pymupdf.readthedocs.io)	CPU only; rendering not required for plain text	Must pass the same gold-sample contract; scanned pages need OCR
Tesseract and OCRmyPDF	Text and searchable PDF; 100+ languages (tesseract-ocr.github.io)	Apache 2.0 engine (tesseract-ocr.github.io); OCRmyPDF MPL-2.0	One process per core (tesseract-ocr.github.io); memory scales with jobs	Needs about 300 DPI (tesseract-ocr.github.io); no structured tables
PaddleOCR pipeline	Detection plus recognition text; orientation and unwarping options	Apache 2.0 code	CPU or GPU	Module timings exclude pre- and post-processing
olmOCR	Markdown from PDF, PNG, JPEG	Apache 2.0 stated on the olmOCR-2 card (huggingface.co)	12 GB GPU RAM minimum for tested setup	Renders every page; author cost claim is conditional
Surya and Marker	Full-page OCR and layout; GPU, CPU or MPS for Marker	Weights licensed separately, free below a threshold	H100 figures at 192 DPI	Commercial weight terms need review
Amazon Textract	Text; separate Forms, Tables, Queries features	Managed cloud; opt-out available	No hardware to run; 3,000-page async limit	Language limits by feature
Other managed APIs	Document OCR, layout, prebuilt models	Provider terms apply	No hardware to run	Prices are estimates

The matrix shows why a single "price per page" misleads. The cheapest path to plain text (CPU extraction) cannot produce table structure from a scan, the pipelines that can produce Markdown need a GPU and a CPU rendering pool, and the managed API removes hardware but carries its own language, size and data-handling limits. Licensing sits on a separate axis again: engine code, model weights and API terms are three different contracts.

Performance and Benchmarks

What published evidence can and cannot support

No public benchmark known to this report measures the cost per accepted page of these engines on an arbitrary enterprise archive, and none should be expected to. The available evidence is useful for designing a test, not replacing one.

An October 18, 2026 study compared cloud and self-hosted OCR on text recognition across 315 pages and nine document types (arxiv.org), with a Gemini-generated reference transcription (arxiv.org). Its authors state that the reference must not be treated as evidence of faithful transcription (arxiv.org), that the mathematical subsets are small (arxiv.org), and that runtime was measured one page at a time and may not reflect batched throughput (arxiv.org). It reports that Surya had the lowest mean character error rate among self-hosted systems, on a single RTX A6000 with 48 GB. It also found that near-perfect page scores could hide a replaced Arabic variable. The study therefore supports uncertainty-aware test design, and does not justify an enterprise accuracy guarantee.

Benchmark designers have also moved away from single scores. OmniDocBench scores reading order with normalized edit distance (arxiv.org) and argues that generic text-similarity metrics fail to fairly assess formulas and tables (arxiv.org). The olmOCR-Bench authors note that edit distance can partially reward severe reading-order failures (arxiv.org), and include a robustness test for long repeated n-grams (arxiv.org). A separate 2026 study of historical documents defines a hallucination as generated text that lacks direct visual evidence in the source (arxiv.org) and observes that character and word error rates treat all errors as equivalent (arxiv.org).

A minimum acceptance test

An evidence-based acceptance design for the archive combines the following checks, all computed on a human-reviewed gold sample drawn from every stratum:

- **Character error rate (CER):** minimum edit operations normalized by reference length, as in the ISRI method (tesseract-ocr.github.io), reported per stratum and not only on average. Also report the share of pages above a failure threshold, because ISRI-style accuracy can go negative when an engine emits excess text (link.springer.com).
- **Reading order:** normalized edit distance over text blocks for multi-column pages.
- **Table structure:** cell-level or structure-aware matching, since text similarity understates structural errors.
- **Critical fields:** exact-match checks on labeled fields, with a separate pass rate.
- **Notation and numerals:** targeted tests for digits, scripts and formula symbols.
- **Hallucination and repetition:** automated checks for repeated n-grams and text absent from the page image, plus reviewer spot checks.
- **Confidence intervals:** bootstrap or equivalent intervals per stratum so a small sample is not mistaken for a guarantee.

Resolution policy belongs in the same protocol. The GPO study found that downsampling did not improve recognition, contrary to claims that 300 DPI is optimal (www.GovInfo.gov), so the render DPI chosen for throughput must be validated against accuracy, not assumed. Surya's own throughput guidance shows the trade-off in the other direction.

Measuring the whole pipeline

Throughput is the count of contract-qualified outputs divided by end-to-end wall time, with retries counted as occupied resources. The measurement should record, as separate lines: PDF rendering, preprocessing, model loading (cold start), steady-state inference, retries, validation and serialization, and peak CPU, RAM, GPU memory and storage bandwidth. Cold start and steady state must not be blended. The 12% retry rate that the olmOCR authors measured is a reminder that retries consume capacity that a nominal tokens-per-second figure does not show.

Data Analysis and Evidence

Dated price ledger

Table 2 lists the list prices used in this report, with the source date or limit that applies. Prices were retrieved on November 2, 2026; source-page dates are shown as stated, and some pages carry no date, and each must be re-confirmed for the selected region, tier and terms before quoting. Items the research could not obtain are shown as buyer fields.

Item	Listed price	Source date and basis	Calculation on 1,000,000 pages	Caveat
Textract basic text detection	$0.0015 per page, then $0.0006 per page above one million	Worked examples, US West (Oregon), no page date	$1,500 (first tier)	Basic text only
Textract forms plus tables	$0.05 plus $0.015 per page, first tier	Same page and region	$65,000 (first tier)	Output includes OCR
Google Document AI Enterprise OCR	$1.50 per 1,000 pages	Page undated	About $1,500	Free first 1,000; no region stated
Azure Document Intelligence Read	$1.50 per 1,000 pages to 1M	Page undated	$1,500	Layout and prebuilt $10 per 1,000
Mistral OCR 4.1	$4 per 1,000 pages	Model card dated July 16, 2026	$4,000	Annotated pages priced higher
Runpod L40S pod	$1.09 per hour (Secure Cloud); $0.79 per hour (Community Cloud)	Updated October 28, 2026	$795.70 per 730-hour month at the Secure rate	The $1.09 rate is the one shown in the visible pods table; both tiers appear in the page's structured data; security terms unconfirmed
Runpod L4 pod	$0.59 per hour (www.runpod.io)	Updated October 28, 2026	$430.70 per 730-hour month	Tier and security terms unconfirmed

Item	Listed price	Source date and basis	Calculation on 1,000,000 pages	Caveat
Owned server (capex, support, power)	Buyer quote	Not obtained	Buyer field	Include reserve and refresh
CPU instance for rendering	Buyer quote	Not obtained	Buyer field	AWS on-demand tables not machine-readable in the retrieved page (aws.amazon.com)

Table 2 shows that list prices span more than an order of magnitude only because outputs differ: the basic text rate and the forms-plus-tables rate for the same service differ by a factor of about 43 ($0.065 divided by $0.0015). A comparison of a self-hosted plain-text pipeline with the forms-plus-tables rate, or of a self-hosted structured parser with the basic rate, would produce a misleading ratio in either direction. Runpod's page also advertises reserved capacity and contract pricing through sales (www.runpod.io) and a footprint of 30+ regions (www.runpod.io), so the displayed hourly rate is a starting point and not a quote.

Capacity: the deadline calculation

For a 1,000,000-page backlog, the required accepted-page rate is simple arithmetic: 2,976 pages per hour over 336 hours (24 hours per day for 14 days), 3,571 per hour over 280 hours (20 hours per day), 7,143 per hour over 140 hours (10 hours per day for 14 days) and 10,000 per hour over 100 hours (10 hours per day for 10 working days). The replica count is the ceiling of backlog divided by the product of the measured accepted rate per replica-hour and the available hours, which must then be checked against scheduling, retry queues and an explicit fault reserve.

Table 3 applies that formula to placeholder per-replica rates. **These rates are scenario inputs, not measurements.** The 727 pages per hour row is the compute-only parity rate against the basic Textract rate ($1.09 divided by $0.0015). The 4,501 row is the L40S rate implied by the olmOCR paper's 1,288-page test in 17 minutes 10 seconds, transferred by arithmetic only.

Accepted pages per replica-hour (placeholder)	GPU-hours for backlog	Compute at $1.09/h (occupied-hours floor)	Replicas: 336 h window	280 h window	140 h window	100 h window
500	2,000	$2,180	6	8	15	20
727 (parity with $0.0015)	1,376	$1,500	5	5	10	14
1,000	1,000	$1,090	3	4	8	10
2,000	500	$545	2	2	4	5
4,501 (paper-implied)	222	$242	1	1	2	3

Table 3 reads as a sensitivity surface. Because compute cost is inversely proportional to accepted rate, a pipeline at half the rate costs twice as much, and the window length changes the number of replicas but not the occupied-hour floor. Billed hours can exceed occupied hours when rented replicas sit idle or wait on CPU

rendering, so the compute column is a lower bound. At the parity row, a pipeline that must also pass through a CPU rendering pool, validation and a retry queue may need more replicas than the GPU alone suggests. Against the forms-plus-tables rate, parity is about 17 accepted pages per replica-hour; against Mistral's $4 per 1,000 it is about 273; against Azure layout at $10 per 1,000 it is 109. Those low thresholds explain why the decision is rarely made on compute price alone.

Steady intake and idle policy

At 100,000 pages per month, the average arrival rate is about 137 pages per hour over a 730-hour month. A dedicated L40S listing billed continuously costs $795.70 per month, or $0.00796 per page at 100,000 pages, which is 5.3 times the $150 basic Textract bill for the same volume. If a pipeline processed 1,000 accepted pages per replica-hour and ran only when batches arrived, the same month needs about 100 GPU-hours, or $109. Utilization, not the hourly rate, drives that difference. The $795.70 versus 530,467-page parity is arithmetic on listed prices alone and is not a break-even, because it omits throughput, quality, idle policy, staff, reserve and output equivalence.

An existing private GPU server changes the accounting but not the logic. Its cost is the marginal expense plus the opportunity cost of whatever online inference or training is displaced, not zero. Whether the ingestion burst can run in off-peak windows without raising latency for online workloads is a scheduling test to run, not assume.

Workbook fields to label

A procurement workbook built on these formulas should expose each input with a unit, a source and a date. At minimum:

- **Backlog and deadline:** pages, calendar days, available hours per day and working days.
- **Steady intake:** pages per month and peak-week multiplier.
- **Accepted pages per replica-hour:** measured per stratum, with retries counted as occupied time.
- **Fault reserve:** extra replicas or hours, stated as a number and not a percentage hidden in a rate.
- **Rental or API price:** tier, region, bundle, date and quote reference.
- **Owned server quote:** capex, support, power, cooling, rack and refresh cycle (buyer fields, unknown here).
- **Review parameters:** reviewed share, seconds per page and loaded hourly rate.
- **License and support lines:** engine, weights, API terms and any commercial fees.

Review labor and accepted-output cost

Full cost per accepted page is compute plus persistence and egress plus license and support plus operations plus review and rework, divided by accepted outputs. Review cost equals reviewed pages multiplied by observed review seconds per page, divided by 3,600, multiplied by the loaded hourly labor rate. The U.S. Bureau of Labor Statistics publishes occupational wages; its May 2023 national table for data entry keyers (a historical release; check the latest OEWS release before use) is one unloaded reference point for a review role (www.bls.gov), and the loaded rate must come from the buyer. The $30 per hour used below is a placeholder.

At $30 per hour, review cost per page of the whole archive is:

- **1% reviewed:** $0.00125 at 15 seconds, $0.0025 at 30 seconds, $0.005 at 60 seconds.
- **5% reviewed:** $0.00625, $0.0125 and $0.025 respectively.
- **10% reviewed:** $0.0125, $0.025 and $0.05 respectively.

Reviewing 5% of 1,000,000 pages at 30 seconds each is 417 reviewer-hours, or $12,500, about eight times the $1,500 basic Textract bill. With six productive hours per person per day over 14 days (84 hours each), that is five reviewers before reserve; over 10 working days it is seven. These are labeled scenario inputs.

Worked comparison (Hypothetical Example)

All inputs here are invented for illustration. Pipeline A is a managed API at $0.0015 per page with 8% of pages routed to review. Pipeline B is a rented L40S at $1.09 per hour processing 2,000 accepted pages per replica-hour, with 12% routed to review because its structure checks flag more pages. Review is 30 seconds per page at $30 per hour. Per 1,000 pages, A costs $1.50 compute plus $20.00 review, or $21.50. B costs $0.545 compute plus $30.00 review, or $30.55. The cheaper engine by compute is the more expensive path by accepted output, because the four-point difference in review share is worth $10 per 1,000 pages while the compute difference is under $1. The reverse outcome occurs if B's review share is lower. The point is to measure the review share, not to adopt these numbers.

Document-level acceptance

If a document is accepted only when all required pages pass, page-level error rates compound. Under an independence assumption, which real scans rarely satisfy and which therefore only illustrates the direction, a 1% page failure rate leaves 90.4% of 10-page documents and 60.5% of 50-page documents intact; at 2% the figures are 81.7% and 36.4%; at 5%, 59.9% and 7.7%. Errors in real archives cluster by scan quality and layout, so the observed document pass rate on the gold sample should be reported directly as cost per accepted document.

Implications and Future Directions

Choosing a path

The evidence in this report supports a staged decision rather than a threshold. No procurement threshold is mandated by any source consulted.

- **Retain CPU or API:** appropriate when the pipeline passes the contract, volume is modest and data terms are acceptable. Basic text at 100,000 pages per month is $150 on the AWS example rate, small beside likely review and engineering cost.
- **Pilot:** run two or three candidate pipelines on the gold sample, with pinned checkpoints, and measure accepted pages per hour, retries and review share per stratum.
- **Rent transient capacity:** if the pilot shows that the backlog fits in a bounded number of GPU-hours, a short rental avoids paying for idle time. Confirm cloud tier, region, hardware bundle, security terms and availability first.

- **Buy:** consider an owned server only after the steady intake, the privacy boundary and the displaced-workload accounting all point to sustained occupancy, using vendor quotes with power, support and refresh. Those quote fields were not obtained for this report.

What would change the answer

Three developments would alter the arithmetic. Quantized checkpoints and newer inference servers raise accepted pages per hour; the olmOCR card recommends the FP8 version for practical use (huggingface.co). Hourly GPU prices also differ by tier: Runpod lists the L40S at $0.79 per hour on Community Cloud and $1.09 per hour on Secure Cloud, a difference of about 38%, so the tier chosen changes the compute line. Evaluation practice is also moving toward uncertainty estimates, structural checks and notation-specific tests.

Electricity is published monthly by the U.S. Energy Information Administration, with 2026 values flagged preliminary (www.eia.gov). At the low power envelope NVIDIA lists for the L4, power is unlikely to dominate the ledger, but buyers should enter their local tariff.

Frequently Asked Questions (FAQs)

What is the AWS Textract price per page?

On the pricing page's US West (Oregon) examples, basic text detection is $0.0015 per page for the first million pages a month and $0.0006 after that. Forms and tables are separate features, with the example using $0.05 and $0.015 per page. Other regions require a check on the live page.

How does self-hosted OCR compare with Textract on cost?

Only at equal output, region and acceptance rule. At $1.09 per hour, GPU compute alone equals the basic Textract rate at about 727 accepted pages per replica-hour, but that excludes storage, staff, reserve, licensing and review. The 530,467-page figure is a fixed-bill parity and not a break-even.

Is a GPU cheaper than a CPU for OCR per page?

It depends on the engine and the output. CPU engines such as Tesseract run one process per core (tesseract-ocr.github.io) and need no GPU, while VLM pipelines such as olmOCR require a GPU. Cost per page is the hourly price divided by measured accepted pages per hour for each, plus review.

What does cost per accepted page mean?

Total cost of compute, persistence, licensing, operations and review, divided by the outputs that pass the acceptance test. Where a document passes only if every page passes, report cost per accepted document as well.

How does open-source OCR compare with Textract on accuracy and cost?

No single current figure applies. A 2021 peer-reviewed comparison ranked Textract second and Tesseract last (link.springer.com), while a 2026 study found Surya, a self-hosted OCR system, lowest among self-hosted systems but scored against a model-generated reference. Run the gold sample.

Can OCR run privately without the cloud?

Yes, with local engines such as Tesseract, PaddleOCR or olmOCR on owned or dedicated hardware. A managed API may store and use inputs unless opted out, and a rented GPU is still a third-party environment whose security terms must be confirmed.

What GPU does self-hosted OCR need?

For olmOCR, a recent NVIDIA GPU with at least 12 GB of GPU RAM was the tested minimum. PaddleOCR-VL's vLLM route is not recommended on T4 or V100-class cards (www.paddleocr.ai). Verify the exact pipeline on the target accelerator.

Conclusion

Private OCR cost per page cannot be read from a price list. The published rates are simple: $0.0015 per basic Textract page in the cited Oregon example, $0.065 for forms plus tables, and $1.09 per hour for a listed L40S on October 28, 2026. The decision-relevant quantity is the full cost of the contract-qualified output divided by accepted pages or documents, which depends on measurements this report does not have.

The staged approach follows from that. Route born-digital pages through CPU text extraction when they pass the same tests. Build a human-reviewed gold sample across every stratum. Measure whole-pipeline throughput with retries and cold start separated. Compute replicas from the deadline, then add reserve. Compare options only at equal output, and include review labor before declaring a GPU or an API cheaper. The hypothesis that correction labor and document mix can reverse a GPU compute advantage is plausible from the arithmetic shown, and it is untested here. All prices should be re-verified at quotation time, and license and data-handling terms reviewed with qualified counsel.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. https://aws.amazon.com/textract/pricing
2. https://www.runpod.io/pricing
3. https://arxiv.org/pdf/2502.18443
4. https://github.com/allenai/olmocr
5. https://arxiv.org/html/2610.25235v1
6. https://tesseract-ocr.github.io/docs/AT-1995.pdf
7. https://arxiv.org/html/2412.07626v2
8. https://github.com/jsvine/pdfplumber
9. https://github.com/pdfminer/pdfminer.six
10. https://github.com/pypdfium2-team/pypdfium2
11. https://pymupdf.readthedocs.io/en/latest/page.html
12. https://ocrmypdf.readthedocs.io/en/latest/advanced.html

13. https://raw.githubusercontent.com/allenai/olmocr/main/olmocr/pipeline.py
14. https://tesseract-ocr.github.io/tessdoc
15. https://tesseract-ocr.github.io/tessdoc/FAQ.html
16. https://tesseract-ocr.github.io/tessdoc/TesseractOpenCL.html
17. https://github.com/ocrmypdf/OCRmyPDF
18. https://www.paddleocr.ai/latest/en/version3.x/pipeline_usage/OCR.html
19. https://link.springer.com/article/10.1007/s42001-021-00149-1
20. https://tesseract-ocr.github.io/tessdoc/ImproveQuality.html
21. https://www.govinfo.gov/media/WhitePaper-OptimizingOCRAccuracy.pdf
22. https://ocrmypdf.readthedocs.io/en/latest/performance.html
23. https://pymupdf.readthedocs.io/en/latest/about.html
24. https://raw.githubusercontent.com/allenai/olmocr/main/olmocr/data/renderpdf.py
25. https://huggingface.co/allenai/olmOCR-2-7B-1025
26. https://huggingface.co/allenai/olmOCR-7B-0725
27. https://github.com/PaddlePaddle/PaddleOCR
28. https://www.paddleocr.ai/latest/en/version3.x/pipeline_usage/PaddleOCR-VL.html
29. https://github.com/datalab-to/surya
30. https://github.com/datalab-to/marker
31. https://www.nvidia.com/en-us/data-center/l4
32. https://aws.amazon.com/ec2/instance-types/g6e
33. https://aws.amazon.com/textract/faqs
34. https://docs.aws.amazon.com/textract/latest/dg/limits-document.md
35. https://aws.amazon.com/service-terms
36. https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_policies_ai-opt-out.md
37. https://cloud.google.com/document-ai/pricing
38. https://azure.microsoft.com/en-us/pricing/details/ai-document-intelligence
39. https://docs.mistral.ai/models/ocr-4-1
40. https://arxiv.org/html/2510.19817v1
41. https://arxiv.org/pdf/2607.24077
42. https://aws.amazon.com/ec2/pricing/on-demand
43. https://www.bls.gov/oes/2023/may/oes439021.htm
44. https://www.eia.gov/electricity/monthly/epm_table_grapher.php?t=epmt_5_6_a

THE PUBLISHER

About GPU Smith

GPU Smith provides independent engineering and research for private AI infrastructure. We help technical buyers and operators reason about GPU workload sizing, hardware selection, procurement, deployment and inference operations, from the compute system to the networking, power and cooling requirements around it.

Start with the workload

Useful infrastructure decisions begin with the models, data, throughput, latency and operating constraints a team actually has. GPU Smith helps connect those requirements with system design and deployment choices. The goal is a privately controlled AI environment whose capacity and operational demands are understood before a purchasing decision.

Hardware and supplier research

Our public reference library covers GPUs, complete systems and networking components. The server-vendor directory and manufacturing research help teams investigate suppliers, compare options and follow sources behind technical claims. We distinguish manufacturer specifications from measured benchmarks and advertised capabilities from tested configurations. Reference pages are research resources, not a live inventory listing or a binding equipment quote.

Deployment and operations

GPU Smith's engineering scope includes infrastructure integration, networking, power, cooling and the practical operation of inference workloads. Our published articles explain the tradeoffs behind deployment, procurement and ongoing operations so teams can ask better questions and document decisions.

Work with GPU Smith

Explore the hardware library, GPU references, networking references and vendor research. Contact GPU Smith with your workloads and deployment constraints to discuss a project and confirm current scope, pricing and availability.

Inclusion of a manufacturer or product in our research does not imply a partnership, certification or endorsement.

Website: https://gpusmith.com

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.