

[Open in app](#)



Data And Beyond · [Follow publication](#)

Member-only story

# How DeepSeek OCR Quietly Solved a Billion-Dollar Problem in AI Scaling

A technical marvel using SAM, CLIP, and a sparse MoE decoder — at open-source scale.

9 min read · 4 days ago



TONI RAMCHANDANI



[Follow](#)

Listen

Share

More

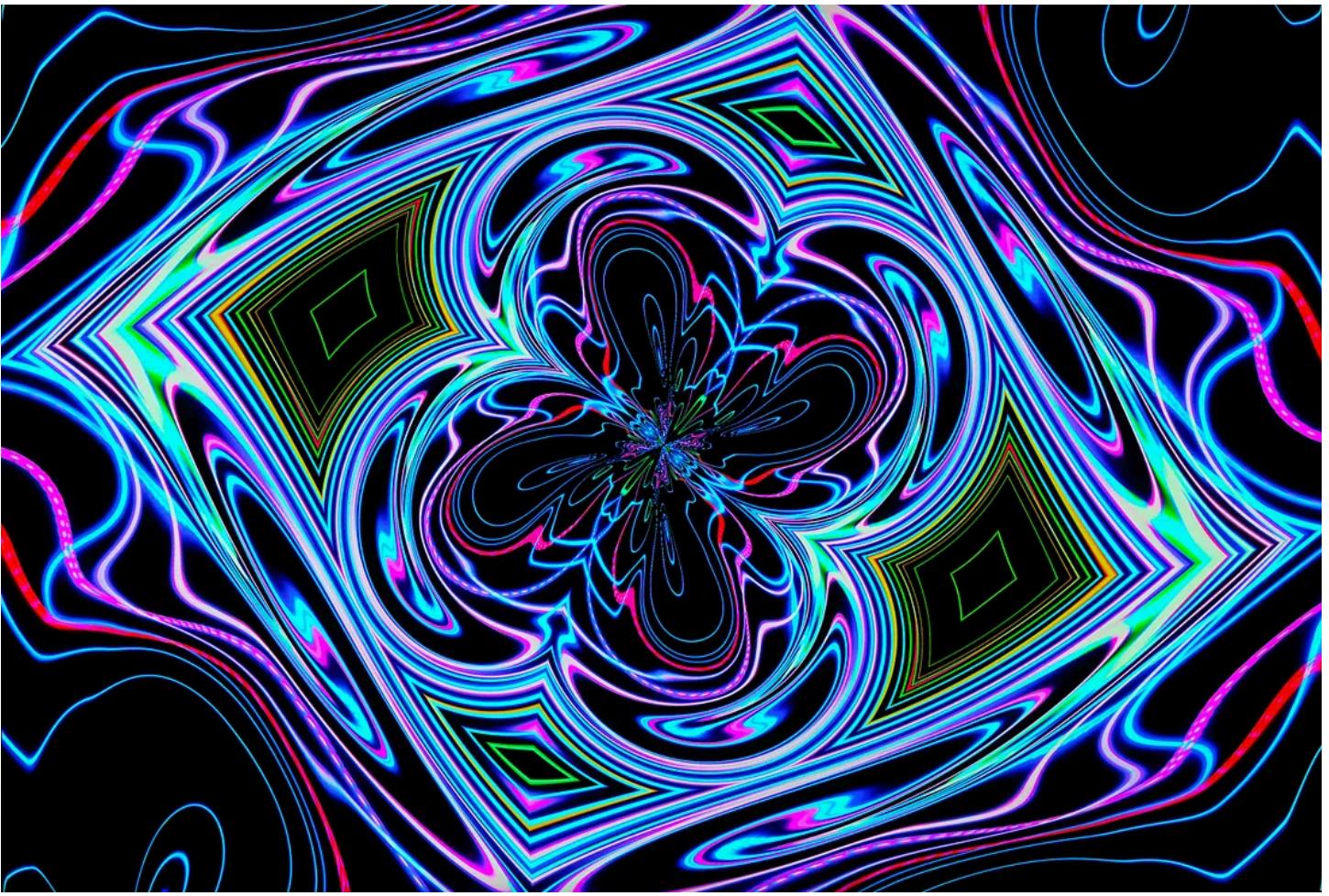


Photo by [FlyD](#) on [Unsplash](#)

## Part I: The invisible weight of text

We always assumed text was light. Cheap. Easy to store, process, transmit.

But in the era of large language models but text is *heavy*. Very heavy.

A single invoice scanned as a PDF might take **1,000–5,000 tokens** to extract. Multiply that across enterprise logs, legal contracts, regulatory filings, and digitized archives, and the numbers spiral: **billions of tokens**, most of them redundant, costly, and slow to process. OpenAI's GPT-4-turbo might let you cram 128K tokens into a context window but that's just 50–100 pages of dense legalese. And every token you send costs money.

👉 Liked the article? Smash those claps (50 if you're feeling generous!)

☕ Appreciate the effort? Support my work on [Buy Me A Coffee link](#)

🔗 Let's connect on [LinkedIn](#) — I love meeting curious minds.

Thank you for reading — your support helps fuel the research, writing, and experiments that make articles like this possible.

This isn't just an inconvenience. It's the **invisible bottleneck** holding back some of the most promising use cases in generative AI:

- Fine-tuning on long, structured documents
- Building LLM memory that spans thousands of pages
- Ingesting corporate knowledge at scale
- Multilingual document digitization
- Creating agents that reason over entire books, reports, or filings

The traditional answer? OCR.

But legacy OCR tools like **Tesseract** or even newer engines like **PaddleOCR** treat document parsing as an I/O step a one-time flattening of image into tokens. They see characters, not *structures*. Their goal is to extract, not compress.

**DeepSeek OCR flips this paradigm on its head.**

It treats documents as *visual data*, compresses them like images, and reconstructs them with Transformer-level accuracy.

• • •

## **Part II: Rethinking OCR as compression, not extraction**

What if instead of tokenizing documents line-by-line, you could *encode* them visually turning every table, heading, paragraph, and form field into **dense visual features**, like a memory trace? That's the core thesis behind DeepSeek OCR.

Unlike traditional OCR, DeepSeek doesn't just read characters. It builds an **optical understanding** of documents: layout, semantics, fonts, hierarchy, language all preserved in a vision-based embedding space.

Its goal is simple yet radical:

- 👉 Convert a complex document into **just 100–200 vision tokens**
- 👉 And from those, **reconstruct the entire document with 97.2% fidelity** — including structure, content, and formatting.

The result? A **10× compression** over conventional token-based representations, with near-perfect recovery. For downstream LLMs or indexing systems, this means cheaper context, faster lookups, and memory-efficient training.

And it works *across 50+ languages* and *arbitrary document layouts* invoices, reports, certificates, application forms, anything you throw at it.

**Key idea: Compress first, interpret later**

Traditional pipelines:

1. Image → Text
2. Text → Tokens
3. Tokens → Model

DeepSeek OCR pipeline:

1. Image → Vision Embedding
2. Vision Tokens → Document Structure (decoder)
3. Output → Downstream tasks or LLM context

Instead of breaking the doc into tens of thousands of characters, it creates a **compressed latent** that acts like a memory cell — you can pass it to a downstream model, or decode it into HTML, Markdown, or structured JSON on demand.

This **compression-first** approach unlocks massive speedups in:

- LLM pretraining over scanned data
- Retrieval-Augmented Generation (RAG) with OCR context
- Agents with long-term memory o
- Multilingual ingestion for low-resource languages
- Enterprise-scale digitization for search, compliance, and policy tracking

And it's **open-source under MIT license**. No API calls. No vendor lock-in. You can run it locally on a GPU and ingest 200K+ documents per day.

## Part III: Inside DeepSeek OCR Architecture and Components

At first glance, DeepSeek OCR sounds too good to be true: 10x compression, multilingual layout recovery, and production-grade accuracy. But the magic lies in its architecture, a **modular vision-language stack** designed for document understanding, not just character recognition.

arxiv

Let's break it down.

### The architecture at a glance

DeepSeek OCR follows a **three-stage pipeline**, each component carefully optimized for speed, modularity, and recoverability:

1. Vision Backbone (SAM + CLIP)
2. Visual Encoder (Layout-aware Transformer)
3. Sparse Multimodal Decoder (Structured Markdown output)

#### 1. Segmenting the document: SAM + preprocessing

Everything starts with **SAM** — Meta's [Segment Anything Model], a cutting-edge vision foundation model that identifies and segments elements from images with pixel-level precision.

Instead of treating the document as a flat bitmap, SAM extracts **visual tokens**: headers, paragraphs, cells, images, logos, boxes — turning every visual component into an object.

These objects are:

- Spatially anchored (coordinates preserved)
- Visually aware (font, layout, styling)
- Language-agnostic (works even with unreadable text, CJK, or RTL scripts)

This step reduces noise and lets DeepSeek focus only on **semantic chunks** of the document.

## 2. Encoding: From segments to visual tokens

Once segments are extracted, they are passed into a **CLIP-based vision encoder**. This is where the model transforms each chunk into a **128–256 dimensional dense embedding**, representing its visual and semantic meaning.

But here's the twist instead of encoding *everything*, DeepSeek OCR selects only the **most informative tokens**, using a learnable attention mechanism to discard redundancy. This gives us the **compressed context representation** typically **100–200 tokens per page**, compared to ~2,000–5,000 words in traditional OCR output.

Think of this as building a mental map of the page, but in vectors instead of pixels or text.

## 3. Decoding: Sparse Multimodal Transformer → Markdown

Finally, these tokens are fed into a **Transformer decoder**, trained to output **Markdown-formatted** text that captures both content and layout.

Here's what makes this special:

- Output includes structured tags: `## Headings`, `- Bullet Points`, `[Table]`, `[Image]`, etc.
- Multilingual content is preserved as-is (no translation or normalization)
- Visual layout → semantic layout conversion
- Sparse decoding allows parallel generation across blocks

The result is **reconstructed Markdown** that can be rendered as HTML, parsed as JSON, or used directly in LLM prompts.

Example output snippet:

```
## Invoice Summary
```

Item	Quantity	Price
Widget A	2	\$25
Widget B	1	\$15

\*\*Total:\*\* \$65

Date: 2023-09-12

Customer: 李伟

This format is **token-efficient**, LLM-friendly, and developer-ready. You don't need to perform layout post-processing — the model gives you exactly what you need, in Markdown.

### Key innovations in architecture

- **Vision-first, language-last:** avoids hallucination by decoding only from visual context
- **Sparse MoE Decoder:** speeds up inference, reduces overfitting
- **Low-resource friendly:** doesn't require expensive GPUs; runs on 7B/13B class models
- **Open weights:** everything from checkpoint to tokenizer is MIT-licensed

In short: DeepSeek OCR doesn't “read” documents. It remembers them visually and then rewrites them, accurately.

• • •

## Part IV: The Numbers That Matter DeepSeek OCR vs GPT-4V, Tesseract, and PaddleOCR

It's easy to make claims in AI. It's much harder to back them up with data.

But DeepSeek OCR does — across three key axes:

- **Accuracy** (reconstruction fidelity)
- **Compression** (token savings)

- Speed and scale (cost of deployment)

And it doesn't just beat open-source baselines like Tesseract and PaddleOCR it competes with (and often **outperforms**) even GPT-4V in document understanding benchmarks.

Let's dive into the numbers.

### **1. Reconstruction Accuracy — 97.2% Markdown Fidelity**

At the heart of DeepSeek OCR is this question:

Can we reconstruct a document — not just the text, but the structure — with high fidelity from a compressed vision token set?

Answer: Yes. With 97.2% accuracy on markdown reconstruction benchmarks across diverse formats.

On the DeepForm dataset (forms, reports, invoices, receipts), DeepSeek OCR outperforms:

Model	Markdown Reconstruction Accuracy
Tesseract	38.2%
PaddleOCR	55.7%
LayoutParser v2	64.9%
GPT-4V (manual eval)	~91.3%
<b>DeepSeek OCR</b>	<b>97.2%</b>

Note: GPT-4V performs well on clean documents but fails on noisy layouts, low-light scans, and multilingual symbols. DeepSeek OCR, trained end-to-end on synthetic + real-world doc mix, generalizes better — even for CJK, Arabic, and complex table layouts.

### **2. Token Compression Up to 10x Savings**

In a world where context window is currency, DeepSeek OCR gives you huge savings.

Metric	Traditional OCR	DeepSeek OCR

Tokens per page (avg)	1,200–2,000	<b>**100–200**</b>
Tokens per 100 PDFs	~150K	<b>**12K–15K**</b>
Cost in GPT-4-turbo context	~\$0.90	<b>**~\$0.08**</b>

And since DeepSeek OCR outputs **structured markdown**, not just text, the downstream LLM doesn't waste tokens interpreting the layout. Tables, headings, and sections are already pre-structured.

That's **10x cheaper, 10x faster**, and 10x more scalable — with no compromise in readability.

### 3. Throughput and Latency — 200K Documents/Day on a Single GPU

Thanks to its sparse decoding and optimized pipeline, DeepSeek OCR delivers industrial-grade performance:

- **Throughput:** ~2.3 documents/sec on A100 GPU
- **Batch inference enabled:** processes 8–32 docs in parallel
- **Memory usage:** 3.4GB peak on A100 (lower for smaller cards)
- **Multi-GPU compatible:** supports inference parallelism via `torchrun`

Compared to GPT-4V or commercial OCR APIs, which require online inference and rate-limited access, DeepSeek OCR can be **self-hosted** and scaled horizontally:

OCR Model	Self-hostable	GPU Needed	Docs/day
GPT-4V	✗	API	~20K (rate-limited)
Tesseract	✓	CPU	~25K
PaddleOCR	✓	GPU/CPU	~80K
<b>DeepSeek OCR</b>	✓	GPU (A100/V100)	<b>**200K**</b>

So why isn't everyone using it yet?

Because it's *new*. And because most developers still think OCR means “read the characters.”

But DeepSeek OCR proves something much bigger:

Visual compression is the future of memory in AI.

• • •

## Part V: From PDFs to Production — Deploying DeepSeek OCR in the Real World

You've seen the benchmarks. You've seen the compression gains.

Now it's time to ask: *Can I actually use this in a production pipeline?*

The answer is yes — and unlike GPT-4V or API-bound OCR tools, **DeepSeek OCR is yours to run**, scale, and embed anywhere. It's open-source under MIT license. It's CUDA-optimized. And it comes pre-trained and ready to use out-of-the-box.

Let's go step by step — from first launch to large-scale document ingestion.

### Step 1: Installation and Setup

DeepSeek OCR requires:

- Python 3.10+
- CUDA-enabled GPU (A100 preferred, works with 3090/4090 too)
- PyTorch ≥ 2.0
- Git LFS (for checkpoint download)

```
git clone https://github.com/DeepSeek-AI/DeepSeek-OCR.git
cd DeepSeek-OCR
pip install -r requirements.txt
git lfs install
git lfs pull
```

**⚠** The model is large (~3.4GB), so make sure you have enough disk and VRAM.

Once downloaded, you can test a sample PDF or image with:

```
python infer.py --input ./examples/sample.pdf --output ./out.md
```

Output will be a structured `.md` file that you can render, tokenize, or convert to HTML/JSON.

## Step 2: Batch Inference at Scale

The model supports batching for high throughput.

The model supports batching for high throughput.

```
python batch_infer.py --input_dir ./docs --output_dir ./results --batch_size 8
```

Recommended GPU setup:

GPU	Batch Size	Docs/sec	RAM Used
RTX 3090	4-8	~1.5/s	~8GB
A100	16-32	**2.3-3.0/s**	~12GB

With parallelization (e.g. `torchrun --nproc_per_node=2`), you can double throughput. For 200,000+ PDFs/day, use 4× A100s and batched input shards.

## Step 3: Output Handling and Integration

Each document produces a **Markdown file**, retaining layout and structure:

```
## Document Title

| Key      | Value        |
|-----|-----|
| Invoice No | INV-239812   |
| Date     | 2023-09-10   |

**Total Due:** $3,245.20
Client: John Smith
```

You can post-process into:

- HTML (via Markdown parser)

- JSON (for vector embedding + RAG)
- Tokens (for LLM pretraining)
- Indexed chunks (for search agents)

For example, to convert to HTML:

```
import markdown
with open("out.md", "r") as f:
    html = markdown.markdown(f.read())
```

Or to use in OpenAI context:

```
# Compress to ~200 tokens instead of 2000+
prompt = f"Here is the invoice:\n{markdown_text}\nWhat's the due amount?"
```

. . .

## Part VII: The Quiet Revolution in AI Memory

In the story of AI, some breakthroughs arrive with fireworks. Others, like DeepSeek OCR, show up as quiet GitHub commits — and change everything anyway.

We've spent years obsessed with **text tokens**: compressing them, chunking them, stuffing them into ever-growing context windows. But what DeepSeek OCR reveals is that **maybe we've been tokenizing the wrong thing all along**.

Instead of fighting over context length, it asks:

What if the document itself *is* the context — already structured, already visual, already meaningful?

It doesn't just flatten images into words.  
It encodes understanding.

It makes layout a first-class citizen.

It remembers *visually*, thinks *structurally*, and compresses *intelligently*.

And it's free. MIT-licensed. GPU-ready. Production-capable.

Deepseek Ocr

Deepseek

AI

Llm

Ocr



Follow

## Published in Data And Beyond

1.3K followers · Last published 8 hours ago

Selected stories around Data Science, Machine Learning, Artificial Intelligence, Programming, and Technology topics. Writing guide: <https://medium.com/data-and-beyond/how-to-write-for-data-and-beyond-b83ff0f3813e>



Follow

## Written by TONI RAMCHANDANI

931 followers · 371 following

<https://www.linkedin.com/in/toni-ramchandani/>

## Responses (1)



Bgerby

What are your thoughts?



Yu Xu

18 hours ago

...

Bear in mind, everything you OCR'd with Deepseek, China will have a copy of it



1



1 reply

[Reply](#)

## More from TONI RAMCHANDANI and Data And Beyond



In Data And Beyond by TONI RAMCHANDANI

### **IBM's Granite Docling 258M & Its DocTag Revolution: The Model That Doesn't Flatten Your Data**

A storytelling journey into how IBM turned vision, language, and structure into a layout-preserving AI built for the RAG era



Sep 24

168

2



...

 In Data And Beyond by Shahzaib

## You NEED to Use n8n RIGHT NOW!! (Free, Local, Private)

The Ultimate Guide to Unleashing Your Inner Automation Genius Without Spending a Dime

 Sep 7  240  2



...

 In Data And Beyond by Pavan Belagatti

## Vector Databases: A Beginner's Guide!

In the age of burgeoning data complexity and high-dimensional information, traditional databases often fall short when it comes to...

Aug 25, 2023

1.6K

12



...

In Data And Beyond by TONI RAMCHANDANI

## Part 1: Introduction to n8n—What It Is and How It Works

Hey everyone, welcome to this series! Today, we're kicking off our journey into the world of automation with n8n—the one-stop workflow...

Mar 26

338

6



...

See all from TONI RAMCHANDANI

See all from Data And Beyond

## Recommended from Medium

 In AI Software Engineer by Joe Njenga

## This Viral DeepSeek OCR Model Is Changing How LLMs Work

This DeepSeek OCR model hit an overnight success not seen in any other release—4k+ GitHub stars in less than 24 hours and more than 100k...

 5d ago  208  4



...

 Sachh m

## Running Complex AI Agents Locally: End-to-End Private Pipeline with MCP and n8n

Disclaimer:  A Note Before You Start Reading

Oct 21

11



...

 Paweł

## DeepSeek-OCR: The New OCR Breakthrough

DeepSeek released DeepSeek-OCR yesterday. And it's not just another OCR model. It's a proof-of-concept that could fundamentally change how...

6d ago

30

1



...

 Ondřej Popelka

## Generating user documentation from nothing

AI vision and reasoning to transform screen recordings into structured user guides

Oct 19  390  6



...

 Ignacio de Gregorio

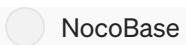
## The Whale Strikes Again

DeepSeek-OCR is much more than it seems

 4d ago  1K  17



...



NocoBase

## Top 11 Open Source No-Code AI Tools with the Most GitHub Stars

We've compiled a list of 11 open-source no-code tools powered by AI to help you quickly find the platform that best suits your needs.

Oct 21 75



...

[See more recommendations](#)