Search models, datasets, users...

← Back to Articles

ColPali: Efficient Document Retrieval with Vision Language Models 👀

Community Article Published July 5, 2024

▲ Upvote 186 +180

manu
Manuel Faysse

Using Vision LLMs + late interaction to improve document retrieval (RAG, search engines, etc.), solely using the image representation of document pages (paper)!

Context

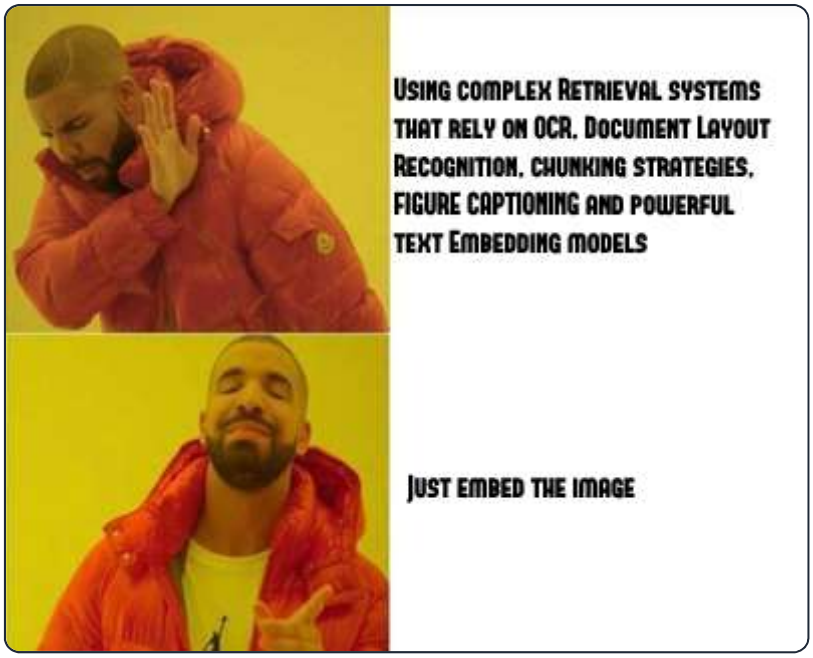
To improve the query answering capabilities of LLMs, it is often best to first search for information online or in external document sets (PDFs), before letting a LLM synthetize a grounded response (RAG). In practice, these retrieval pipelines for PDF documents have a huge impact on performance but are non-trivial...

1. Run Optical Character Recognition (OCR) on scanned PDFs
2. Run Document Layout Detection models to segment pages into paragraphs, figures, titles
3. Reconstruct the structure and the reading order of the page
4. Optionally, use resource intensive specialized models to caption figures, images and tables in natural language
5. Use a chunking strategy to split or merge text passages in a coherent way

6. Use a strong neural embedding model (BGE M3) to map text chunks to a semantically meaningful vector space
7. Store the vector index to be used for future retrieval

Although tools exist to facilitate this pipeline (Unstructured, Surya), the whole indexing process can be slow, tends to propagate errors, and struggles to take into account the more visual elements of a page (tables, figures, images but also fonts, etc..).

Our concept ? Just embed the page image directly !



In practice, it's not as easy as we just made it sound ! Our method, ColPali is enabled by the latest advances in Vision Language Models, notably the PaliGemma model from the Google Zürich team, and leverages multi-vector retrieval through late interaction mechanisms as proposed in ColBERT by Omar Khattab.

Let's break it down, with more technical details !

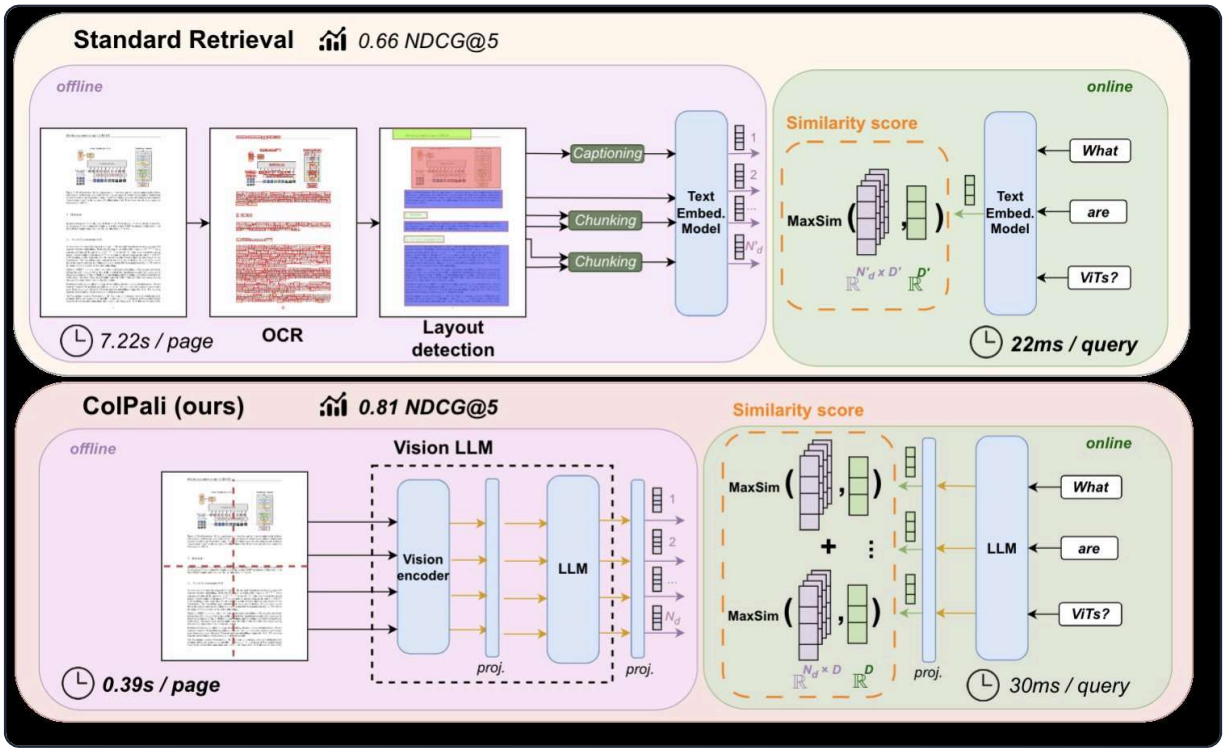
Model Architecture

Many retrieval systems can be broken down into two parts.

- In the indexing phase, all the documents from the corpus are indexed in an offline fashion.
- In the querying phase, a user query is matched with a low latency to the pre-computed document index.

Important requirements for efficient retrieval systems are thus (R1) good retrieving performance, (R2) reasonable indexing speeds, (R3) low latency during querying.

During indexing, standard “bi-encoder” neural retrieval systems first parse documents to extract semantically coherent text passages, then map them to a dense vector space that aims to represent the text’s semantic meaning, and store the resulting “embeddings”. During querying, the query is converted into its dense vector representation and the document passage vectors with the biggest cosine similarity can be retrieved with a low latency.



Our method ColPali is a bit different !

During indexing, we aim to strip away a lot of the complexity by using images (“screenshots”) of the document pages directly.

A Vision LLM (PaliGemma-3B) encodes the image by splitting it into a series of patches, which are fed to a vision transformer (SigLIP-So400m). These patch embeddings are linearly projected and inputted as "soft" tokens to a language model (Gemma 2B), in order to obtain high-quality contextualized patch embeddings in the language model space, which we then project to a lower dimension (D=128) for more efficient storage. We thus construct and store a multi-vector document representation for each page image.

During runtime querying, a user query is embedded by the language model, to obtain token embeddings. We are able to run a ColBERT-style "late interaction" (LI) operation to efficiently match query tokens to document patches. To compute a LI(query, document) score, for each term in the query, we search for the document patch that has the most similar ColPali representation. We then sum the scores of the most similar patches for all terms of the query, to obtain the final query-document score. Intuitively, this late-interaction operation allows for a rich interaction between all terms of the query and document patches, all the while benefiting from the fast matching and offline computation offloading that more standard (bi-encoder) embedding models enable.

> With ColPali, we thus benefit from fast indexing speeds (R2) without significantly impacting querying latencies (R3) ! But what about performance (R1)?

ViDoRe

Although awesome benchmarks exist to evaluate text embedding models, we find that in many practical use cases, the prior document ingestion pipeline matters much more than the embedding model itself ! While documents often rely on visual elements to more efficiently convey information to human readers, text-only systems barely tap into these visual cues. To our knowledge, no benchmark evaluates document retrieval methods by considering both textual and visual document features like a human would.

To this end, we introduce ViDoRe, the Visual Document Retrieval Benchmark, to assess retrievers on their capacity to retrieve visually rich information in docs, with tasks spanning various topics, modalities (figures, tables, text), and languages !

Dataset	# Queries	Domain
Academic Tasks		
DocVQA (eng)	500 (500)	Industrial
InfoVQA (eng)	500 (500)	Infographics
TAT-DQA (eng)	1600 (1600)	Varied Modalities
arXiVQA (eng)	500 (500)	Scientific Figures
TabFQuAD (fra)	210 (210)	Tables
Practical Tasks		
Energy (eng)	100 (1000)	Scientific
Government (eng)	100 (1000)	Administrative
Healthcare (eng)	100 (1000)	Medical
AI (eng)	100 (1000)	Scientific
Shift Project (fra)	100 (1000)	Environment

Table 1: *ViDoRe* comprehensively evaluates multimodal retrieval methods. The size of the document corpus is indicated in parentheses.

ViDoRe is linked to a HF leaderboard https://huggingface.co/spaces/vidore/vidore-leaderboard and we hope to see many models trying out this new "Retrieving in Vision Space" paradigm !

Results

Training details

We initialize the Vision Language Model backbone using pretrained weights from PaliGemma and randomly initialize the final projection layer. To facilitate training, we add low-rank adapters to the language model attention weights, as well as the

linear projection layers. Our training dataset is composed of (query, document image) pairs that we sourced from two main streams. On one hand, we repurposed Visual Question Answering datasets and used the original question as our query, and the associated image as the gold label. To increase the coverage and diversity of the training set, we also collected tens of thousands of permissively licensed PDF documents covering a broad range of topics, and synthetically create relevant queries using the powerful Claude Sonnet Vision model. In total, we gather around 100k pairs, and finetune our model with an in-batch contrastive loss, by attempting to maximize the difference between the matching score of the correct (page, query) pair, and the score of the incorrect pairs.

ColPali results

On ViDoRe, ColPali outperforms all other evaluated systems, including baselines where a very strong proprietary Vision model (Claude Sonnet) is used to caption all visual elements !

	ArxivQ	DocQ	InfoQ	TabF	TATQ	Shift	AI	Energy	Gov.	Health.	Avg.
Unstructured Text only											
- BM25	-	34.1	-	-	44.0	59.6	90.4	78.3	78.8	82.6	-
- BGE-M3	-	28.4↓5.7	-	-	36.1↓7.9	68.5↑8.9	88.4↓2.0	76.8↓1.5	77.7↓1.1	84.6↑2.0	-
Unstructured + OCR											
- BM25	31.6	36.8	62.9	46.5	62.7	64.3	92.8	85.9	83.9	87.2	65.5
- BGE-M3	31.4↓0.2	25.7↓11.1	60.1↓2.8	70.8↑24.3	50.5↓12.2	**73.2**↑8.9	90.2↓2.6	83.6↓2.3	84.9↑1.0	91.1↑3.9	66.1↑0.6
Unstructured + Captioning											
- BM25	40.1	38.4	70.0	35.4	61.5	60.9	88.0	84.7	82.7	89.2	65.1
- BGE-M3	35.7↓4.4	32.9↓5.4	71.9↑1.9	69.1↑33.7	43.8↓17.7	73.1↑12.2	88.8↑0.8	83.3↓1.4	80.4↓2.3	91.3↑2.1	67.0↑1.9
Contrastive VLMs											
Jina-CLIP	25.4	11.9	35.5	20.2	3.3	3.8	15.2	19.7	21.4	20.8	17.7
Nomic-vision	17.1	10.7	30.1	16.3	2.7	1.1	12.9	10.9	11.4	15.7	12.9
SigLIP (Vanilla)	43.2	30.3	64.1	58.1	26.2	18.7	62.5	65.7	66.1	79.1	51.4
Ours											
SigLIP (Vanilla)	43.2	30.3	64.1	58.1	26.2	18.7	62.5	65.7	66.1	79.1	51.4
BiSigLIP (+fine-tuning)	58.5↑15.3	32.9↑2.6	70.5↑6.4	62.7↑4.6	30.5↑4.3	26.5↑7.8	74.3↑11.8	73.7↑8.0	74.2↑8.1	82.3↑3.2	58.6↑7.2
BiPali (+LLM)	56.5↓-2.0	30.0↓-2.9	67.4↓-3.1	76.9↑14.2	33.4↑2.9	43.7↑17.2	71.2↓-3.1	61.9↓-11.7	73.8↓-0.4	73.6↓-8.8	58.8↑0.2
ColPali (+Late Inter.)	**79.1**↑22.6	**54.4**↑24.5	**81.8**↑14.4	**83.9**↑7.0	**65.8**↑32.4	**73.2**↑29.5	**96.2**↑25.0	**91.0**↑29.1	**92.7**↑18.9	**94.4**↑20.8	**81.3**↑22.5

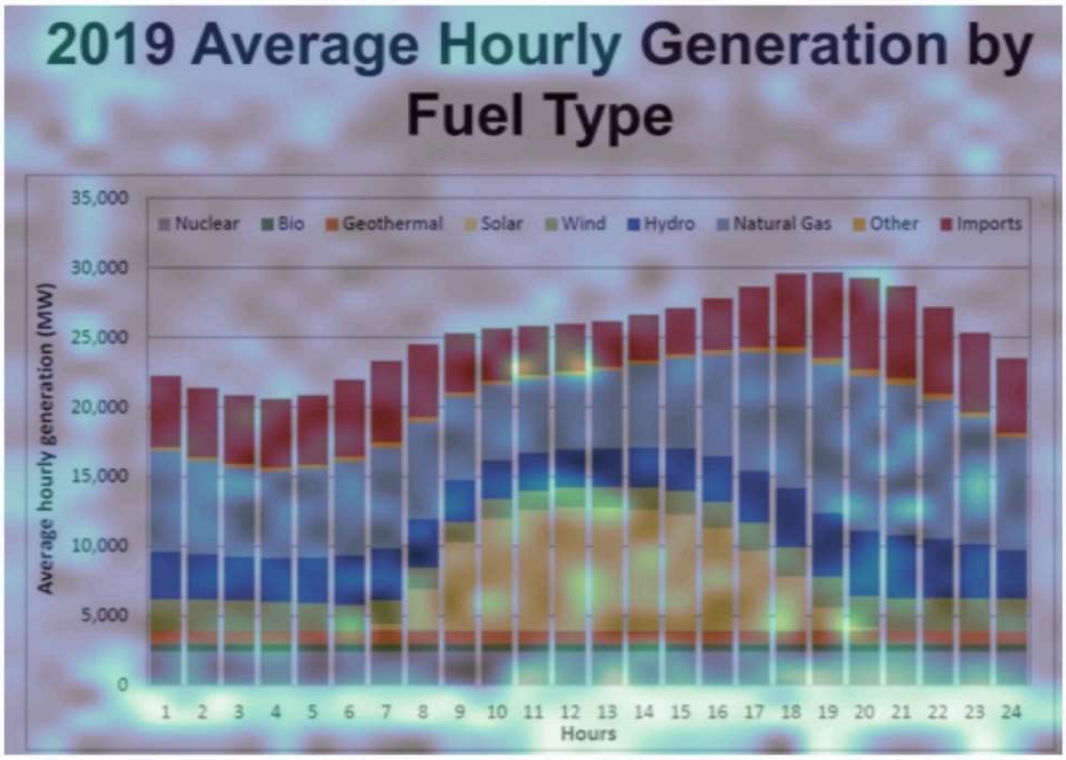
Table 2: **Comprehensive evaluation of baseline models and our proposed method on *ViDoRe*.** Results are presented using NDCG@5 metrics, and illustrate the impact of different components. Text-only metrics are not computed for benchmarks with only visual elements.

The difference is particularly stark on the more visually complex benchmark tasks, such as InfographicVQA, ArxivQA, and TabFQuAD representing respectively infographics, figures, and tables. However, text centric documents are also better

retrieved by the ColPali model across all evaluated domains and languages, making our approach the overall best performing document-retrieval model on ViDoRe !

Interpretability

Beyond speed and performance, another interesting feature of ColPali, is that it enables visualizing which patches of a document stand out w.r.t. a given query. Here the term `<hour>` matches patches containing words like "hourly" but also the x-axis representing time, showcasing good chart comprehension !



Query: "Which hour of the day had the highest overall eletricity generation in 2019?"

Figure 1: For each term in a user query, **ColPali** identifies the most relevant document image patches (highlighted zones) and computes a query-to-page matching score. We can then swiftly retrieve the most relevant documents from a large pre-indexed corpus.

Conclusion

This blogpost is already long enough but good news, tons more resources, informations and ablations exist, and will keep on coming !

📝 The paper: https://arxiv.org/abs/2407.01449

💾 The benchmark: https://huggingface.co/vidore

👀 The model: https://huggingface.co/vidore/colpali

💻 The benchmark code: https://github.com/illuin-tech/vidore-benchmark

💻 The training code: https://github.com/ManuelFay/colpali

❌ X of the first authors: @ManuelFaysse, @sibille_hugues, @tonywu_71

Citation

```
@misc{faysse2024colpaliefficientdocumentretrieval,
      title={ColPali: Efficient Document Retrieval with Vision Languag
      author={Manuel Faysse and Hugues Sibille and Tony Wu and Bilel O
      year={2024},
      eprint={2407.01449},
      archivePrefix={arXiv},
      primaryClass={cs.IR},
      url={https://arxiv.org/abs/2407.01449},
}
```

Acknowledgments

This work primarly stems from an academic-industry partnership between CentraleSupélec and Illuin Technology, with involvement from actors at Equall.ai and ETH Zürich. It benefits from a compute grant from CINES ADASTRA (Grant 2024-AD011015443). Joint work by Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, Pierre Colombo.

System theme

Company

TOS

Privacy

About

Jobs

Website

Models

Datasets

Spaces

Pricing

Docs