Multi-modal

LlamaIndex offers capabilities to not only build language-based applications, but also **multi-modal** applications - combining language and images.

Types of Multi-modal Use Cases

This space is actively being explored right now, but there are some fascinating use cases popping up.

Multi-Modal RAG (Retrieval Augmented Generation)

All the core RAG concepts: indexing, retrieval, and synthesis, can be extended into the image setting.

- The input could be text or image.
- The stored knowledge base can consist of text or images.
- The inputs to response generation can be text or image.
- The final response can be text or image.

Check out our guides below:

```{toctree}
---
maxdepth: 1
---
/examples/multi_modal/gpt4v_multi_modal_retrieval.ipynb
Multi-modal retrieval with CLIP </examples/multi_modal/multi_modal_retrieval.ipynb>
Image to Image Retrieval </examples/multi_modal/image_to_image_retrieval.ipynb>
```

Retrieval-Augmented Image Captioning

Oftentimes understanding an image requires looking up information from a knowledge base. A flow here is retrieval-augmented image captioning - first caption the image with a multi-modal model, then refine the caption by retrieving from a text corpus.

Check out our guides below:

```{toctree}
---
maxdepth: 1
---
/examples/multi_modal/llava_multi_modal_tesla_10q.ipynb
```

```

### LLaVa-13, Fuyu-8B and MiniGPT-4 Multi-Modal LLM Models Comparison for Image Reasoning

These notebooks show how to use different Multi-Modal LLM models for image understanding/reasoning. The various model inferences are supported by Replicate or OpenAI GPT4-V API. We compared several popular Multi-Modal LLMs:

- GPT4-V (OpenAI API)
- LLava-13B (Replicate)
- Fuyu-8B (Replicate)
- MiniGPT-4 (Replicate)
- CogVLM (Replicate)

Check out our guides below:

```{toctree}
---
maxdepth: 1
---
/examples/multi_modal/replicate_multi_modal.ipynb
GPT4-V: </examples/multi_modal/openai_multi_modal.ipynb>
```

Pydantic Program for Generating Structured Output for Multi-Modal LLMs

You can generate `structured` output with new OpenAI GPT4V via LlamaIndex. The user just needs to specify a Pydantic object to define the structure of output.

Check out the guide below:

```{toctree}
---
maxdepth: 1
---
/examples/multi_modal/multi_modal_pydantic.ipynb
```

Chain of Thought (COT) Prompting for GPT4-V

GPT4-V has amazed us with its ability to analyze images and even generate website code from visuals.
```

This tutorial investigates GPT4-V's proficiency in interpreting bar charts, scatter plots, and tables. We aim to assess whether specific questioning and chain of thought prompting can yield better responses compared to broader inquiries. Our demonstration seeks to determine if GPT-4V can exceed these known limitations with precise questioning and systematic reasoning techniques.

```{toctree}
---
maxdepth: 1
---
/examples/multi_modal/gpt4v_experiments_cot.ipynb
```

Simple Evaluation of Multi-Modal RAG

In this notebook guide, we'll demonstrate how to evaluate a Multi-Modal RAG system. As in the text-only case, we will consider the evaluation of Retrievers and Generators separately. As we alluded in our blog on the topic of Evaluating Multi-Modal RAGs, our approach here involves the application of adapted versions of the usual techniques for evaluating both Retriever and Generator (used for the text-only case). These adapted versions are part of the llama-index library (i.e., evaluation module), and this notebook will walk you through how you can apply them to your evaluation use-cases.

```{toctree}
---
maxdepth: 1
---
/examples/evaluation/multi_modal/multi_modal_rag_evaluation.ipynb
```

Using Chroma for Multi-Modal retrieval with single vector store

Chroma vector DB supports single vector store for indexing both images and texts.
Check out out Chroma + LlamaIndex integration with single Multi-Modal Vector Store for both images/texts index and retrieval.

```{toctree}
---
maxdepth: 1
---
/examples/multi_modal/ChromaMultiModalDemo.ipynb
```

Multi-Modal RAG on PDF's with Tables using Microsoft `Table Transformer`

One common challenge with RAG (Retrieval-Augmented Generation) involves handling PDFs that contain tables. Parsing tables in various formats can be quite complex.

However, Microsoft's newly released model, Table Transformer, offers a promising solution for detecting tables within images.

In this notebook, we will demonstrate how to leverage the Table Transformer model in conjunction with GPT4-V to yield better results for images containing tables.

The experiment is divided into the following parts and we compared those 4 options for extracting table information from PDFs:

1. Retrieving relevant images (PDF pages) and sending them to GPT4-V to respond to queries.
2. Regarding every PDF page as an image, let GPT4-V do the image reasoning for each page. Build Text Vector Store index for the image reasonings. Query the answer against the `Image Reasoning Vector Store`.
3. Using Table Transformer to crop the table information from the retrieved images and then sending these cropped images to GPT4-V for query responses.
4. Applying OCR on cropped table images and send the data to GPT4/ GPT-3.5 to answer the query.

```{toctree}
---
maxdepth: 1
---
/examples/multi_modal/multi_modal_pdf_tables.ipynb
```