

Candidate Assignment: Multi-Modal Document Intelligence (RAG-Based QA System)

Introduction

With the rapid growth of digital documents, organizations increasingly deal with unstructured and semi-structured data such as PDFs containing text, tables, and images. Traditional document processing systems struggle to extract meaningful information from such heterogeneous content.

This project proposes a Multi-Modal Document Intelligence System that ingests documents containing text, tables, and images, converts them into a unified semantic representation, and enables context-aware question answering using Retrieval-Augmented Generation (RAG). The system allows users to query documents naturally and receive accurate, source-grounded answers.

Problem Statement

Most existing document QA systems:

- Work only on plain text
- Fail on scanned or image-based PDFs
- Lack source attribution
- Are restricted by API quota and rate limits

There is a need for a **scalable, local, and multi-modal solution** capable of:

- Handling diverse document formats
- Extracting knowledge from multiple modalities
- Providing transparent and reliable answers

Objectives

The main objectives of this project are:

- To extract text, tables, and images from structured and scanned PDFs
- To perform OCR on image-based documents
- To convert multi-modal data into a unified embedding space

- To enable semantic document retrieval using vector similarity
- To implement a RAG-based chatbot for document question answering
- To provide page-level source attribution for every response

Key Features

- **Multi-Modal Ingestion**
Extracts text, tables, and images (via OCR) from PDFs
- **Smart Chunking & Embeddings**
Performs semantic and structural document segmentation
- **Vector-Based Retrieval**
Enables efficient similarity search over multi-modal content
- **RAG-Powered QA Chatbot**
Generates accurate responses grounded in retrieved context
- **Source Attribution**
Provides page-level or section-level citations
- **Evaluation Ready**
Supports benchmarking across text, table, and image queries

System Architecture

The system consists of the following components:

1. **Document Loader** – Loads PDFs and identifies document structure
2. **Multi-Modal Extractor** – Extracts text, tables, and images
3. **OCR Engine** – Converts image content into text
4. **Chunking Module** – Segments documents into meaningful units
5. **Embedding Generator** – Converts chunks into vector embeddings
6. **Vector Database** – Stores and retrieves embeddings efficiently
7. **RAG Pipeline** – Combines retrieval and generation for QA
8. **User Interface** – Allows users to ask questions and view answers

Technologies Used

Component	Technology
Programming Language	Python
OCR Engine	Tesseract OCR
Embedding Models	Local LLM Embeddings (via Ollama)
LLM Inference	Ollama (Local Models)
Backend	Python
UI	Streamlit
Document Processing	LangChain / PDF libraries

Why Local Models (Ollama)?

During development, cloud-based APIs caused **rate limit errors and quota restrictions**, which interrupted experimentation and evaluation.

Advantages of Local Models:

- No API key required
- No quota or rate limits
- Fully offline execution
- Better control over data privacy
- Faster iteration during development

This approach ensures system reliability and removes dependency on external services.

Implementation Details

1. PDFs are ingested and parsed into text, tables, and images
2. OCR is applied to scanned pages and embedded images
3. Extracted content is chunked semantically
4. Each chunk is converted into vector embeddings

5. Embeddings are stored in a vector database
6. User queries are embedded and matched using similarity search
7. Relevant context is passed to the LLM for answer generation
8. The final response includes source references

Live Demonstration

A live demo video has been included in the project repository demonstrating:

- Document ingestion
- OCR processing
- Query execution
- RAG-based responses
- Source attribution

Results and Evaluation

The system was tested on:

- Text-heavy PDFs
- Scanned documents
- PDFs containing tables and images

Observations:

- High accuracy for text and table queries
- OCR effectively extracted content from scanned documents
- Source attribution improved answer trustworthiness
- Local model inference performed consistently without failures

Conclusion

This project successfully demonstrates a **robust, scalable, and multi-modal document intelligence system**. By combining OCR, vector retrieval, and RAG, the system enables reliable document understanding and question answering without reliance on external APIs, making it suitable for real-world and enterprise use cases.