# Multimodal Data Management

Course: Algorithms, Data Structures and Databases

Authors:

Edwin Santiago Delgado Rospigliosi

Toni

Project Title: Medical Visual Assistant (Educational)

Date: October 2025

## Introduction and Context

This project was developed as part of the *Algorithms, Data Structures and Databases* course and serves as an applied introduction to the design and implementation of complete data management systems. The primary objective is to explore how core data engineering concepts, including structured data handling, efficient storage, and automated workflows, can be integrated into a cohesive, reproducible pipeline using modern tools and best practices.

The project focuses on a multimodal data management scenario, where textual and visual medical data (e.g., radiology reports and X-ray images) are processed together within a unified system. By constructing this pipeline, students gain practical exposure to the foundational components of real-world data projects: data ingestion, transformation, validation, and exploitation.

Beyond its technical scope, the project introduces essential DataOps principles, encouraging systematic handling of data quality, automation, and version control. Through the integration of tools such as MinIO (object storage), Python (processing and orchestration), and ChromaDB (semantic vector storage), the exercise provides hands-on experience in multimodal data management and AI-driven data exploration.

## Project Objectives

The goal of this project is to design and implement a complete multimodal data management pipeline that integrates text and image data within an educational, reproducible framework. Each objective outlined below corresponds to a key component of the system and reflects both the theoretical and practical aspects of the course.

* **Design a multimodal data management system supporting multiple modalities:** The project combines medical text and image data (radiology notes and X-ray scans) within a unified structure, enabling cross-modal relationships and retrieval. This serves as a practical exercise in designing efficient data structures for diverse data types.
* **Implement a zonal data pipeline structured into four stages (Landing, Formatted, Trusted, Exploitation):** The architecture applies DataOps-inspired principles, where each zone represents a stage of data maturity. Scripts automate ingestion, cleaning, and validation, ensuring data traceability and reproducibility across the workflow.

## Data Context and Sources

**Domain**

The project is situated within the field of medical imaging and clinical data management, an area that increasingly relies on artificial intelligence and data engineering techniques to organize and analyze large volumes of heterogeneous medical information. It focuses specifically on multimodal healthcare data, combining radiological images, such as chest X-rays, with textual medical notes describing patient observations and findings.

This domain is characterized by the coexistence of visual and textual information, both of which contain valuable insights that, when integrated, can enhance clinical understanding and data-driven healthcare decision-making. The project thus serves as an educational exercise in connecting these two complementary modalities under a unified data architecture.

**Problem Statement**

The main problem addressed by this project is the absence of an integrated and automated framework capable of managing and analyzing multimodal medical data. In most clinical and research environments, radiology images and written reports are stored in separate systems with limited interoperability.

As a result, retrieving relevant information or identifying similar cases across modalities requires manual effort and domain expertise. The challenge, therefore, lies in designing a data management pipeline that can systematically ingest, process, and relate image and text data, enabling efficient similarity searches, cross-modal queries, and automatic image captioning.

This pipeline also aims to ensure data quality, reproducibility, and scalability, following DataOps principles such as modularity, transparency, and automation.

**Business Value**

Although this project is conducted for academic purposes, the proposed system has significant practical and professional relevance. The ability to efficiently manage and relate medical images with textual information has direct applications in clinical research, medical education, and healthcare data governance.

A functional multimodal data pipeline can:

* Support healthcare professionals by enabling faster retrieval of similar medical cases and improving access to relevant information.
* Assist students and researchers in building structured, high-quality datasets for experimentation in medical AI.
* Provide healthcare institutions with a foundation for secure, auditable, and efficient data management.

By demonstrating how multimodal data can be harmonized through a structured and automated pipeline, this project illustrates how DataOps methodologies can contribute to more intelligent, efficient, and reproducible medical data systems.

**Data Modalities**

The dataset used in this project contains two primary modalities, medical images and textual descriptions, each providing complementary information within the medical context.

* **Image Modality:** Comprises chest X-ray scans in .png or .jpg format. Each image represents a clinical observation, serving as the visual foundation for similarity search and captioning tasks.
* **Text Modality:** Consists of short, anonymized radiology reports in .txt format. These texts summarize medical findings such as lung opacity, inflammation, or pneumonia-related conditions. The textual data is cleaned, normalized, and stored in a consistent structure for downstream processing.

Together, these modalities allow the system to establish relationships between medical imagery and descriptive text, forming the basis for multimodal retrieval and analysis.

**Data Sources**

The data used for experimentation originates from a combination of public educational datasets and synthetic examples created for demonstration purposes.

* The primary source is the Chest X-Ray Pneumonia dataset (Kaggle), which provides labeled chest radiographs categorized as *normal* or *pneumonia*.
* Supplementary synthetic radiology notes were generated manually to simulate realistic text-image associations, ensuring that the dataset remains fully anonymized and ethically compliant.

This combination results in a small but representative dataset suitable for demonstrating multimodal processing and retrieval without exposing sensitive clinical information.

**Value of the Dataset**

The curated dataset supports the development and validation of the pipeline across all four zones of the architecture. Its design enables:

* **Similarity-based retrieval** — identifying visually or semantically related medical cases.
* **Cross-modal exploration** — allowing text queries (e.g., “lung opacity in right lobe”) to retrieve matching X-rays.
* **Automatic caption generation** — producing descriptive summaries of unseen medical images through generative models.

By uniting both modalities under a controlled educational context, the dataset provides a realistic yet ethically sound foundation for exploring AI-assisted medical data management and practical DataOps integration in multimodal systems.

## Data Management Backbone

**Data Division and Zone Architecture**

The multimodal data pipeline developed for this project follows a zonal architecture based on the DataOps framework, which promotes the separation of data according to its processing state, quality level, and purpose. This approach ensures traceability, reproducibility, and quality assurance throughout the data lifecycle. The project adopts three main stages — the Landing Zone, the Formatted Zone, and the Trusted Zone — before reaching the Exploitation Zone, where the analytical processes take place.

**Landing Zone**

The Landing Zone acts as the initial entry point for all incoming data. It is designed to capture and store raw, unmodified data from different modalities — in this case, medical images and textual reports.

* Each file is stored in MinIO under specific subdirectories for temporal and persistent storage.
* No transformations are applied at this stage, preserving the original format, metadata, and structure for auditing and reproducibility.
* The ingestion scripts (landing/ingest.py) automatically upload new files to the corresponding S3 buckets, providing a clear handoff point between data collection and processing.

This layer ensures that the project maintains a single source of truth, allowing the reproduction of any transformation or analysis later in the pipeline.

**Formatted Zone**

Once the raw data is stored, it moves to the Formatted Zone, where standardization and normalization occur. The goal of this zone is to ensure that all files share a consistent structure and compatible format for further processing.

* Image data is converted into a uniform format (e.g., .png) with consistent naming and size specifications.
* Text files are normalized (encoding, casing, spacing) and stored in a clean, structured form.
* The script (formatted/format\_images\_text.py) ensures reproducibility by applying deterministic transformations.

This stage effectively prepares data for analytical operations, facilitating reliable and uniform downstream processing.

**Trusted Zone**

The Trusted Zone is the most critical quality checkpoint in the pipeline. It guarantees that all data entering the exploitation phase meets a defined quality standard.

* For image data, duplicate and low-resolution files are detected and removed using perceptual hashing and dimension checks.
* For text data, anonymization routines are applied to remove or mask potential identifiers, ensuring ethical handling of medical information.
* Each transformation is logged, and a data quality report (reports/quality\_report.md) is generated, summarizing the number of validated, dropped, and corrected entries.

By enforcing data validation, this zone ensures that only clean, reliable, and ethically compliant data flows into the analytical components of the system.

**Exploitation Zone Overview**

The Exploitation Zone focuses on embedding generation and similarity search. Here, multimodal data is transformed into vector representations using CLIP and stored in ChromaDB, allowing for same-modality, cross-modality, and generative queries.

**Summary of Zones**

|  |  |  |
| --- | --- | --- |
| **Zone** | **Purpose** | **Tools / Techniques** |
| **Landing Zone** | Raw data ingestion and secure storage | Python, MinIO |
| **Formatted Zone** | Format standardization and normalization | Pillow, text preprocessing |
| **Trusted Zone** | Data validation and anonymization | imagehash, regex, logging |
| **Exploitation Zone** | Embedding generation and multimodal search | CLIP, ChromaDB |

This zonal organization not only supports clear data provenance and process transparency but also provides a solid foundation for extending the pipeline into an operational, AI-driven environment in subsequent project stages.

## Implementation Summary

The implementation of the Medical Visual Assistant project is organized around a modular, automated pipeline that handles data from ingestion to exploitation. Each stage is built as a Python component interacting with cloud storage, transformation utilities, and embedding databases, ensuring an end-to-end reproducible workflow.

**5.1 Tools and Environment**

The project integrates several open-source technologies to simulate a realistic data engineering environment:

|  |  |  |
| --- | --- | --- |
| **Component** | **Purpose** | **Technology / Tool** |
| **Storage Layer** | Object storage for raw and processed data | MinIO (Dockerized S3-compatible storage) |
| **Processing Layer** | Data ingestion, transformation, and quality checks | Python 3.11, Pandas, Pillow, imagehash, regex |
| **Embedding and Retrieval Layer** | Multimodal vector representations and semantic search | OpenCLIP (HuggingFace) + ChromaDB |
| **Generative Model** | Image captioning and descriptive text generation | BLIP Model (HuggingFace Transformers) |
| **Automation and Versioning** | Reproducibility, CI/CD, and orchestration | run\_pipeline.py, Docker Compose, GitHub |

All dependencies are containerized using Docker Compose, which launches both the MinIO service and supporting Python environment. The pipeline runs locally but mirrors a production-grade cloud architecture, emphasizing reproducibility and modularity.

**5.2 Data Input and Ingestion (Landing Zone)**

The pipeline begins in the Landing Zone, where raw data — images and text — is collected into structured local directories. The ingestion script (src/landing/ingest.py) automatically scans these folders and uploads all files to the MinIO bucket system. Each file path is preserved to maintain lineage, and MinIO acts as the single source of truth. At this stage:

* **Image files** are stored without transformation.
* **Text files** are uploaded in their raw .txt form.
* A metadata manifest is generated to track filenames, timestamps, and upload status.

**5.3 Data Transformation and Standardization (Formatted Zone)**

The Formatted Zone performs all data cleaning and normalization tasks through the script src/formatted/format\_images\_text.py. Here, the pipeline ensures all data adheres to a consistent structure, which is essential for embedding generation later.

**Key steps:**

* Images:
* Resized and converted to .png format.
* Renamed using a deterministic convention (img\_####.png).
* Invalid or corrupted files are skipped and logged.

**Text:**

* Encodings normalized to UTF-8.
* Punctuation, spacing, and casing standardized.
* Unstructured lines merged into coherent sentences.

A validation check confirms that each text file has a corresponding image before progressing to the Trusted Zone. The result is a clean, homogeneous dataset ready for quality validation and AI processing.

* 1. **Data Validation and Quality Control (Trusted Zone)**

In the Trusted Zone, the script src/trusted/validate\_data.py applies a sequence of validation and cleaning routines:

* **Image Validation:**
  + Detects and removes duplicates using perceptual hashing (imagehash).
  + Verifies minimum resolution thresholds.
  + Records invalid or missing entries.
* **Text Validation:**
  + Applies **regex filters** to identify and remove potential personal identifiers (names, IDs, etc.).
  + Ensures that no file contains empty or unreadable content.
  + Generates a standardized JSON log containing all accepted entries.

The output of this phase is stored in a trusted/ directory, and a data quality report is generated to ensure that only high-quality, anonymized, and reliable data progresses to the exploitation stage.

**5.5 Data Embedding and Retrieval (Exploitation Zone)**

Once validated, the data enters the Exploitation Zone, where embeddings are created and stored for search and generative tasks.

1. **Feature Extraction:**
   * Each image and its corresponding text are processed through **OpenCLIP** to generate 512-dimensional embeddings.
   * These embeddings capture semantic similarity across modalities.
2. **Storage in Vector Database:**
   * All embeddings are stored in ChromaDB, which supports efficient nearest-neighbor search.
   * Metadata (file names, modality type, hash) is attached to each record.
3. **Cross-Modal Search:**
   * Users can query with text (e.g., *“opacity in left lung”*) to retrieve the most visually relevant images.
   * The same infrastructure also supports image → image or text → text searches.
4. **Generative Captioning:**
   * The BLIP model generates natural-language captions for each image, producing descriptive summaries that can be indexed alongside the embeddings.

The result is a fully functional multimodal knowledge base that connects medical images and text in a searchable format.

**5.6 Output and Final Artifacts**

The final outputs of the system include

* **Processed datasets** stored in MinIO, organized by zone.
* **Vector embeddings** in ChromaDB, accessible for similarity search.
* **Generated captions** stored as .txt metadata files.
* **Quality reports and logs** for transparency and auditing.

When fully executed, the pipeline provides an operational multimodal assistant capable of:

* Finding similar X-rays based on visual or textual input.
* Generating automatic descriptive captions.
* Demonstrating an end-to-end DataOps workflow suitable for educational and research applications.

## Analytical Tasks

The analytical component of the project demonstrates how the processed data can be exploited for meaningful outcomes. The first task, same-modality search, focuses on identifying visually similar chest X-ray images using CLIP embeddings stored in ChromaDB. When a new X-ray is provided, its embedding is compared with existing vectors in the database, and the system retrieves images that share the highest semantic similarity. This process showcases the ability of the pipeline to index and search complex visual data efficiently.

The second task, multi-modality search, highlights the system’s capacity to connect different data types. A textual query such as *“lung opacity in right lobe”* is converted into an embedding using the same CLIP model and compared against the image embeddings. The system then returns the most visually relevant X-ray images, demonstrating how text and image data can be linked through a shared semantic representation.

Finally, the third task, generative captioning, explores the interpretability of the dataset. Using the BLIP image captioning model, the system generates concise textual descriptions for X-ray images, simulating an automated reporting tool. This component not only enriches the dataset with synthetic text but also validates the multimodal framework by converting image content back into human-readable form.

## Data Quality Report (Trusted Zone)

Data quality validation is a central feature of the Trusted Zone, where the integrity and consistency of all processed data are verified. During this phase, the pipeline records key metrics, including the total number of images processed, the count of low-resolution or duplicate images removed, and the number of cleaned text files. It also documents the replacement or masking of any potential identifiers to ensure that no sensitive information remains in the dataset.

All validated outputs are standardized into .png and .txt formats, guaranteeing uniformity for later processing. A detailed data quality report is automatically generated, summarizing these results. This report, stored in the reports/ directory, provides a transparent overview of the cleaning operations and ensures the reproducibility of data preparation stages.

## Operations and Automation

The pipeline’s execution and automation are managed by a centralized orchestration script, run\_pipeline.py, which sequentially executes all phases from ingestion to exploitation. This design allows for one-command reproducibility, enabling the full workflow to be redeployed or re-executed at any time with consistent results.

To ensure reliability and maintainability, the project employs a lightweight CI/CD pipeline through GitHub, which automatically checks code formatting and dependency integrity. The entire environment, including MinIO and Python services, is containerized via Docker Compose, ensuring consistent operation across systems. Configuration parameters such as access credentials and directory paths are stored in an .env file, isolating sensitive information from the source code. Deterministic transformations and fixed random seeds further guarantee that embeddings and validation outputs remain reproducible between runs.

## Results

Upon full execution, the pipeline successfully processed the dataset through all four zones — Landing, Formatted, Trusted, and Exploitation. The embeddings were generated using CLIP and stored in ChromaDB, enabling both same-modality and cross-modality retrieval. The BLIP model produced accurate and coherent image captions, validating the integration of the generative component.

Example output from the captioning module included sentences such as *“A grayscale X-ray showing lung opacity consistent with pneumonia”*, which aligns with the expected description of the image content. These results confirm that the system meets its primary objectives: structured multimodal ingestion, reliable data validation, and AI-driven retrieval and generation.

## Conclusions and Next Steps

The Medical Visual Assistant project successfully demonstrates a functional multimodal data management pipeline that integrates textual and visual data within a unified and reproducible structure. By applying DataOps principles — including reproducibility, quality control, automation, and modular design — the project offers a realistic foundation for exploring data workflows in medical or educational contexts.

Future improvements planned for Part 2 include the development of an interactive Streamlit user interface for multimodal querying, the automation of data ingestion through scheduled jobs, and the integration of advanced analytics such as retrieval-augmented generation (RAG) or image classification. Additional monitoring and evaluation dashboards will also be introduced to measure performance and data quality over time, transforming the current prototype into a fully interactive, scalable system.