Documentation

This file provides clear **setup instructions, script execution details, and database access methods**.

DICOM Metadata Extraction & Storage Pipeline

- This project extracts metadata from **DICOM medical images**, stores it in **SQLite**, and generates **summary statistics & visualizations.**

Environment Setup & Dependencies

Install Python Packages
Ensure Python 3.11 is installed. Then install the required libraries using:

bash
pip install -r requirements.txt

Required Packages

The **requirements.txt** includes:

pydicom
pandas
matplotlib
seaborn
sqlite3 # This is built-in, no need to install separately

Running the Scripts

Extract DICOM Metadata

Run the metadata extraction script to generate **dicom_metadata.csv:**

bash

python extract_metadata.py

Store Data in SQLite

Run the database storage script to insert extracted metadata:

bash

python store_metadata.py

Generate Summary Statistics & Visualization

Run the analytics script to compute statistics and show visualizations:

bash

python analyze_metadata.py

Accessing & Viewing the Database

Open SQLite Database

To manually explore the stored DICOM metadata, open SQLite:

bash

sqlite3 dicom_metadata.db

Run SQL queries:

sql

SELECT * FROM dicom_metadata LIMIT 10;

To view the database using **DB Browser for SQLite**, install it from:

https://sqlitebrowser.org/dl/

Schema Definition (DDL Statements)

```
CREATE TABLE IF NOT EXISTS dicom_metadata (
    PatientID TEXT,
    StudyInstanceUID TEXT PRIMARY KEY,
```

```
    SeriesInstanceUID TEXT,
    SliceThickness TEXT,
    PixelSpacing TEXT,
    StudyDate TEXT,
    AcquisitionDate TEXT,
    DICOM_File TEXT,
    NumSlicesPerSeries INTEGER
);
```

Schema Description

- **PatientID**: Unique identifier for the patient.
- **StudyInstanceUID**: Unique ID for a medical study.
- **SeriesInstanceUID**: Unique ID for a DICOM series.
- **SliceThickness**: Thickness of each slice in mm.
- **PixelSpacing**: Distance between pixels in mm.
- **StudyDate**: Date of the study.
- **AcquisitionDate**: Date the images were captured.
- **DICOM_File**: Path to the original DICOM file.
- **NumSlicesPerSeries**: Number of slices in a series.

Summary Statistics of Extracted DICOM Metadata

✔ **Total number of studies**: 10

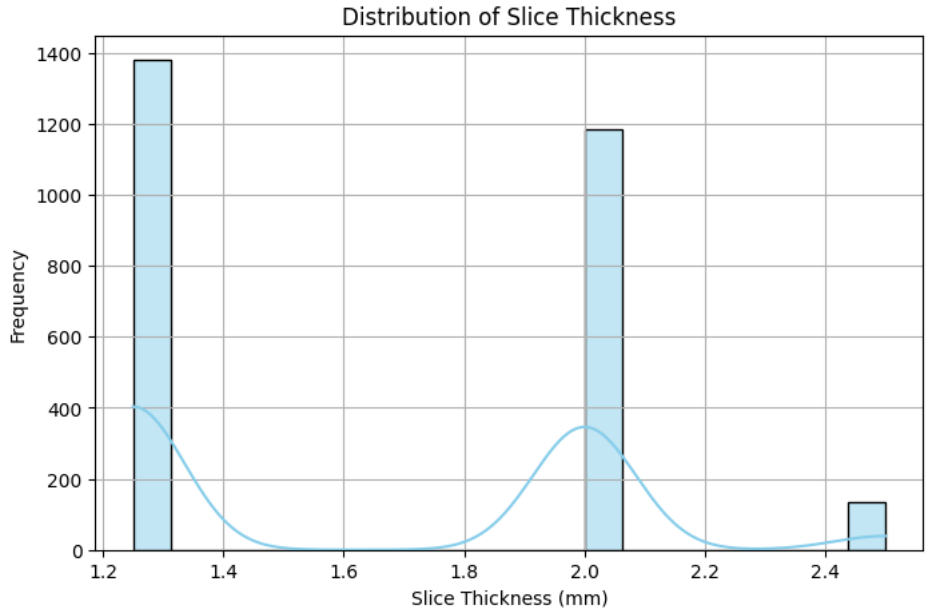
✔ **Total slices across all scans:** 820706

✔ **Average number of slices per study**: 82070.60

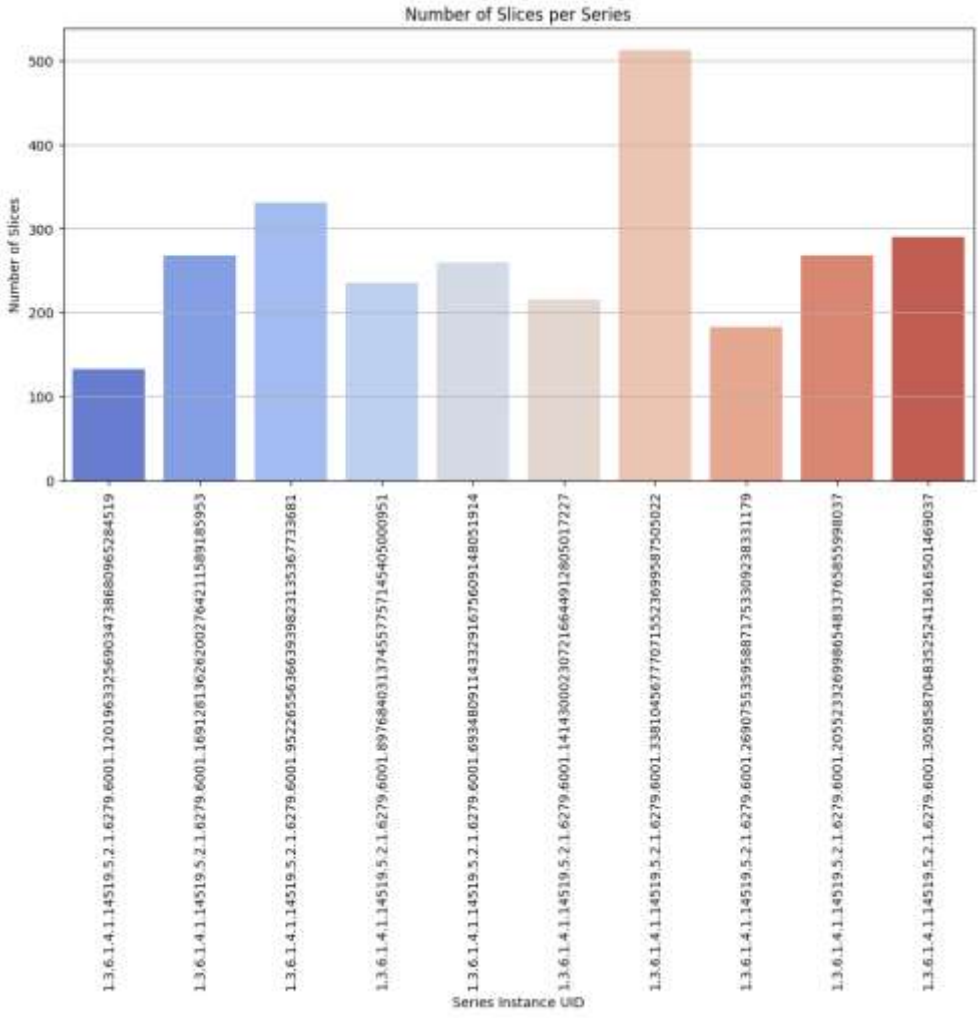
✔ **Most common slice thickness:** 2.5mm

Visualizations

Histogram of Slice Thickness



Bar Chart of Slices per Series



Scalability Note

How to Handle 1,000+ Scans Efficiently

1. **Parallel Processing**
    - Use multiprocessing or Dask to process multiple DICOM files concurrently.
    - Example: **concurrent.futures.ThreadPoolExecutor** for efficient metadata extraction.
2. **Cloud Storage & Databases**
    - Store DICOM files in **AWS S3 / Google Cloud Storage** instead of local storage.
    - Use **PostgreSQL, MongoDB, or Google BigQuery** for metadata instead of SQLite.
3. **Distributed Computing**
    - Use **Apache Spark or Hadoop** for large-scale DICOM metadata extraction.
    - Implement **batch processing pipelines** with Apache Airflow.
4. **Real-time Processing**
    - Stream data using **Apache Kafka or AWS Kinesis** for continuous updates.
    - Set up **auto-scaling clusters (AWS Lambda, Google Cloud Functions)** for on-demand processing.
5. **Monitoring & Logging**
    - **Track Failures**: Log errors using logging module or send alerts via AWS SNS.
    - **Performance Metrics**: Use **Prometheus + Grafana dashboards** for monitoring.
    - **Failure Recovery**: Implement **checkpoints in databases** to resume processing from failures.