EUCAIM WP5 Task Force 2

Tools for data quality assessment and data cleaning

Image Duplicates Checker (AUTH)

Summary: The tool scans one or more directories for DICOM files, organizes them into series, and identifies duplicates at both the file and series level. It uses DICOM metadata (e.g., SOPInstanceUID and SeriesInstanceUID) to detect exact intra- and cross-series duplicates, performs series-level hashing to identify identical image sets, and applies pixel-based similarity metrics such as SSIM, cosine similarity, and MAD to detect near-duplicate series. All results are compiled into a structured report summarizing exact duplicates, near-duplicates, and potential matches.

Status : containerized

Contacts : difoto@auth.gr, ioannach@auth.gr

Table of Contents

I. Purpose

The tool automatically detects and reports duplicate or near-duplicate DICOM image series within large medical imaging datasets. It employs a tiered deduplication pipeline that combines metadata filtering, hash-based matching, and pixel-level similarity analysis. By identifying redundant or duplicated scans across directories, the tool helps ensure the integrity and unbiased composition of AI training datasets, ultimately improving model reliability and reproducibility.

II. Tool description for its conceptual validation

The conceptual validation aims to ensure that a tool is aligned in its purpose and functionality with the pre-processing tools specified in EUCAIM T5.3. For this, the following documentation will detail **the necessity of the tool and its functioning**.

An oral presentation has been given on **24/04/2024** during a focus group meeting dedicated to data quality assessment and data cleaning, for partners to understand the tool better as well as to provide feedback.
Link to the presentation of the tool :
https://docs.google.com/presentation/d/1KrixTi2QD8B23gqkXaY2ru50ijuoZXZi/edit?usp=drive_link&ouid=105979482259582415027&rtpof=true&sd=true
Link to the video of the oral presentation : link

1. Name: **DICOM image similarity/duplicate checker**

2. Contributor: **Aristotle University of Thessaloniki (AUTH)**

3. Area: **Clinical and Imaging Data Quality assessment**

4. Tool description:

The tool implements a multi-stage deduplication pipeline designed for efficient detection of duplicate DICOM series across large datasets. It begins with a parallel metadata scan, grouping images by patient and series identifiers while allowing exclusion of specific modalities (e.g., SEG, RTSTRUCT files).

The pipeline then applies a tiered filtering and comparison strategy:

1. **Metadata-level filtering: Identifies identical or overlapping SOP Instance UIDs within or across series, and constructs candidate pairs only among comparable modalities.**
2. **Series-level hashing: Generates a SHA-based hash signature for each series using representative DICOM slices. This hash acts as a fingerprint, enabling fast detection of identical series while minimizing I/O.**
3. **Pixel-similarity comparison: For non-identical but *similar* series, the tool compares sampled image slices using structural (SSIM), cosine, and mean-absolute-difference metrics to detect *near duplicates*.**

5. Data: a dataset with DICOM series/images

6. Methodology/performance: **The tool is containerized with Docker and will be distributed as a Docker image. Since processing DICOM images can be very resource intensive, we will also consider setting resource limits (e.g. CPU, RAM) during dockerization to avoid excessive consumption of system resources.**

7. Use: **detailed documentation will accompany the tool**

8. Input/output formats: description of the input/output of the tool at the validation stage (will help if some adaptation is needed).

Input : A directory path containing one or more DICOM series (standard .dcm files).

Optional parameters allow exclusion of specific modalities (e.g., SEG, RTSTRUCT) and control over parallelism, sampling rate, and similarity thresholds.

Output : A JSON report summarizing results from all pipeline stages. Specifically:

- **Counts of scanned files, detected duplicates, and compared series.**
- **Metadata- and hash-based duplicate groups.**

- **Pixel-level similarity results for candidate pairs, including per-metric scores (SSIM, cosine, and mean absolute difference) and the overall mean similarity.**

9. Quantitative results: performance obtained during training of the tool (if applies)

   N/A

10. Qualitative results: Provide some visual results (if available) of applying such tools.

    Example of one execution:

```
(general) general difoto@tpad ~/Desktop/annotation-task python dedup_pipeline.py ./ --workers=4
[1] Scanning 2973 DICOM files...
Reading DICOM headers: 100%| | 2973/2973 [00:25<00:00, 116.19file/s]
[2] Filtering series and finding metadata-level duplicates...
[3] Computing series SHA hashes (sample_step = 5)...
Hashing series: 100%| | 27/27 [00:01<00:00, 24.09series/s]
Pixel similarity: 100%| | 317/317 [00:26<00:00, 12.18pair/s]

[DEBUG] Sample pixel-similarity results (top 5 by mean score):
  LUNG1-009 ↔ LUNG1-016  mean=0.838  metrics={'ssim': 0.6911222601093769, 'cosine': 0.8882565259933471, 'mad': 0.935981285572052}
  LUNG1-007 ↔ LUNG1-023  mean=0.819  metrics={'ssim': 0.6754459883935837, 'cosine': 0.8474366188049316, 'mad': 0.933384382724762}
  LUNG1-018 ↔ LUNG1-016  mean=0.819  metrics={'ssim': 0.7015040681819273, 'cosine': 0.8012901544570923, 'mad': 0.9532151579856872}
  LUNG1-004 ↔ LUNG1-016  mean=0.797  metrics={'ssim': 0.6626826699325289, 'cosine': 0.7880999684333801, 'mad': 0.9410661339759827}
  LUNG1-025 ↔ LUNG1-002  mean=0.797  metrics={'ssim': 0.6509163283757653, 'cosine': 0.7939345955848693, 'mad': 0.9456564426422119}
{
  "files_total": 2973,
  "series_total": 27,
  "errors": 0,
  "intra_series_dups": 1,
  "cross_series_dups": 100,
  "candidate_pairs": 318,
  "unique_hashes": 26,
  "hash_duplicate_groups": 1
}

[4] Pixel-similarity summary:
{
  "pairs_tested": 317,
  "near_duplicates": 0
}
```

11. Additional information: successful use cases, external resources (open code, papers...) licence, certification … (if they apply).

    Successful use cases: **N/A**

    Licence: **Custom Software License Agreement**

    Acknowledgements: **No**

    Keywords for searching in databases: **data curation, data quality control, data cleaning, data redundancy, imaging data, duplicate images**

EUCAIM WP5 Task Force 2

Tools for data quality assessment and data cleaning

III. Technical specifications

This part of the documentation is dedicated to providing all relevant technical specifications to prepare for the tool's integration into the EUCAIM test environment.

It is possible that the tool's integration will require, among others, some modifications in the input/output, or the inclusion of monitoring mechanisms.

1. Data:

   All data processed by the tool are DICOM images collected as part of the INCISIVE project and/or data from open repositories (e.g. TCIA). The tool is configured to work exclusively with DICOM image formats, as the pre-filtering steps are based on DICOM-specific metadata. If compatibility with other image formats (e.g. .jpeg, .png) is required, slight changes to the tool are necessary.

2. Methods:

   Pipeline architecture:

   - **Step 1: Parallel metadata scan and series indexing.**
   - **Step 2: Metadata-based filtering and duplicate detection.**
   - **Step 3: Series-level SHA hashing for byte-identical detection.**
   - **Step 4: Pixel-similarity analysis (SSIM, cosine, MAD) for near-duplicate detection.**

   The tool is modular, allowing each stage to be executed independently or as a unified pipeline.

3. Specific Technical information:

   a. CPU: Scales linearly with <*number of workers*>
   b. Programming language : Python (version ≥ 3.10)
   c. Expected RAM usage : 8-16GB
   d. Running mode (interactive/batch-based/case-based…): batch-based
   e. Software version : 2.0
   f. Libraries : pydicom, pandas
   g. Security measures: requires sudo privileges (to run the docker image)

4. Traceability and monitoring mechanism: **No such mechanism has been implemented, but a method for error logging can be integrated**

5. Unitary tests: **description of the tests implemented to verify the correct functioning of the tool.**

   Validation tests were performed to verify the correct functioning of each stage of the DICOM deduplication pipeline.

   Example of test execution with test dataset:

   A subset of DICOM series derived from the LUNG1 collection (TCIA) was used. Controlled duplicates were manually introduced to simulate real-world redundancy scenarios:

   - **A single duplicated slice within an existing CT series.**
   - **A partially duplicated series (some slices copied into another folder).**
   - **A fully duplicated series (entire series duplicated in a different directory).**

6. Access restriction : **Yes, the source code should be protected and not accessible to other partners.**

7. Additional information for tool integration: **tool is configured to run with DICOM image format. If a different image format is required e.g. jpeg slight modifications should be performed**

IV. Integration validation

1. Communication channel for the helpdesk
   a. difoto@auth.gr
   b. loannach@auth.gr

2. Most common errors
   a. **Tool has not been tested officially.**

3. FAQs
   a. N/A

4. User Manual
    a. Installation/configuration instructions : **a readme with instructions will be provided**

    b. Usage instructions : **a readme with instructions will be provided**

    c. Additional considerations: **none**

5. **Integration tests**: Description of tests for assessing the correct integration of the tool

    A test with a standard dataset that will be provided, along with the expected output.

6. Foreseen issues

The main challenges in deploying the DICOM Deduplication Tool on a centralized platform concern resource usage and data consistency.

- **Resource utilization: Comparing large datasets across multiple repositories can be CPU- and memory-intensive. We address this with parallel processing, tiered filtering, and series-level hashing,**
- **Data consistency: DICOM data from different partners may vary in formatting, metadata completeness, or resolution. The tool includes error handling, and configurable thresholds to handle such heterogeneity and avoid false matches.**
- **Integration considerations: Deployment on shared infrastructures (e.g., EUCAIM) may require adapting file paths, security permissions, etc.. Feedback on specific problematic cases is welcome to refine pre-processing and ensure consistent, reliable deduplication across datasets.**