

Python-based Quality Assessment Tools for Medical Imaging

This manuscript ([permalink](#)) was automatically generated from mirzask/QA_paper@b55609c on May 16, 2020.

Authors

- **Riqiang Gao**

 [XXXX-XXXX-XXXX-XXXX](#) ·  [RiqiangGao](#) ·  [johndoe](#)

Electrical Engineering and Computer Science, Vanderbilt University · Funded by Grant XXXXXXXXX

- **Mirza S. Khan**

 [0000-0001-7007-9437](#) ·  [mirzask](#)

U.S. Department of Veterans Affairs, Tennessee Valley Healthcare System; Department of Biomedical Informatics, Vanderbilt University

Abstract

Ensuring good quality of medical images (e.g., computed tomography, CT) is essential for achieve maximum diagnostic information and image interpretation. While in practice, medical imaging can be destroyed by many factors such as system error at data acquisition and network transferring. Multiple existed tools are developed to deal with medical imaging, while most of them are either fail to handle multiple errors may encounter or provide comprehensive quality assessment (QA) report. In this paper, we develop a whole pipeline with several open source applications to assess the quality of CT images from the original DICOM file, and convert to 3D NIFTI files. Seven possible steps are checked to make sure the medical images are appropriate for next steps (diagnostic usage or machine learning). Our QA tool has been proved to be useful and time-saving for clinical laboratory usage and processing data for image interpretation.

The code and tutorials are publicly available at https://github.com/MASILab/QA_tool.

Introduction

The primary use case for medical imaging is for interpretation by radiologists and other trained persons. Secondary use of medical imaging for research and discovery has become more prominent and has led to many meaningful contributions to imaging and medicine.

Often, analyses relying on medical imaging utilize the raw Digital Imaging and Communications in Medicine (DICOM) image files or other common image file formats, such as nrrd or NIfTI. In medical image de-identification, minor encoding errors or manipulation of DICOM file data could render these files useless or inaccessible [citation]. On a similar note, adjustments/modifications to image files, such as those used to aid a radiologists' view, may pose challenges for image modeling tasks. Yet, ~~there remains little consensus or tools providing a pipeline~~ little attention has been afforded to ensure the quality of medical image files prior to use in model training ~~to be used in downstream analyses~~.

Data quality is often an under-appreciated but vital component for machine learning tasks, including natural language processing and computer vision. This is often described by the adage, "garbage in, garbage out." In order to address this matter for our own model development, we have developed a series of sensible, automated data quality checks for medical imaging data.

Here we present our tool to assist with quality assessment of medical image files... We demonstrate its use on NLST and x other imaging studies... (some NLST image files are of poor quality; how does this affect model performance?). Demonstrate improved performance of downstream learning tasks by using satisfactory images for training.

Intuition and Task Description

In this paper, we describe the several steps of quality assessment (QA) for the de-identified CT scans, including the steps converting 2D DICOM files to 3D NIfTI.

Related Tools used in our Pipeline

Detailed Steps

Instance Number Checking

Instance Number (IN) is a number that identifies this image in DICOM files, which was named Image Number in earlier versions [1]. Generally, the INs of DICOM images in a single scan should be a series of consecutive integers. Our first priority is to check if there are any missing INs of the CT scan. We check the IN using

$$C_1 = \max \{in_i\} - \min \{in_i\} + 1 - \text{size} \{in_i\}$$
$$C_2 = \sum_{i,j, i < j} 1 \{in_i = in_j\}$$

where $\{in_i\}$ is the list of INs of the DICOM files, and $\text{size} \{in_i\}$ indicates how many DICOMs in the scan. C_1 and C_2 represent how many slices are missed in the scan and how many slice-pairs with the same IN, respectively. The CT scan passes the Instance Number check if $C_1 = 0$ and $C_2 = 0$.

Slice Distance Checking

Slice Location is defined as the relative position of the image plane expressed in mm. This information is relative to an unspecified implementation specific reference point [1]. Slice Distance (SD) is defined as subtraction of Slice Location of the two consecutive DICOMs. Generally speaking, the SD of all consecutive pairs should be the same in a scan, while in practice, the system error might result in very small difference. Herein, we introduce a self-defined threshold to tolerate the system error. The SD checking is

$$C_3 = \sum_i 1 \{sd_i < \varepsilon\}$$

where $\{sd_i\}$ is the list of SDs of DICOM files, ε is the self-defined tolerance threshold. C_3 represents how many places which slice distance error. The CT scan pass the Instance Number checking if $C_3 = 0$.

Filtering Scans with few slices

Some scans with unreasonable limited number of slices but can easily pass the Instance Number checking and Slice Distance checking. For example, if a chest scan only with 3 DICOM slices, it cannot be usable. Filtering those scans can be easily done with a self-defined threshold on size $\{in_i\}$.

$$C_4 = 1 \{\text{size } \{in_i\} < \delta\}$$

where δ is the self-defined threshold for filtering scans with few slices. The CT scan pass the filtering checking if $C_4 = 0$.

Physical Length Filtering

Some scans are unreasonable extended out of the Region of Interest. For example, the target scan is chest CT while is given a whole-body scan. Another situation is only a chunk of chest CT is given without a complete lung. The *Physical Length Filtering* is designed for selecting those CTs with unreasonable physical body length for further processing or removing. Two thresholds are needed (i.e., low bound and upper bound of physical body length) for filtering

$$C_5 = 1 \{\sigma_1 < PL < \sigma_2\}$$

where PL is the physical length computed from Slice Location of DICOMs, σ_1 and σ_2 are the self-defined low bound and upper bound. The CT scan pass the filtering checking if $C_5 = 1$.

Convert DICOM files to NifTI

We use an open-source tool `dcm2niix` [2] to convert DICOMs file to NIFTI.

NifTI orientation Check and Resolution filtering

We check the CT orientation and resolution in the affine matrix A , whose size is 4x4. We have

$$C_6 = 1 \{-A_{11} = A_{22} = A_{33} > 0\}$$

$$C_7 = \sum_{i=1}^3 1 \{|A_{ii}| > \Phi_i\}$$

where Φ is the thresholds of 3 dimensions. The CT scan is not with standard orientation if $C_6 = 0$, and we use the open source `fslreorient2std` to convert CT to standard orientation. $C_7 > 0$

represents that the CT does not match resolution requirements.

Double check with visualized slices

We use the `slicesdir` [citation] to visualize scans.

Methods

Results



Figure 1: Case of Instance Number check failure.

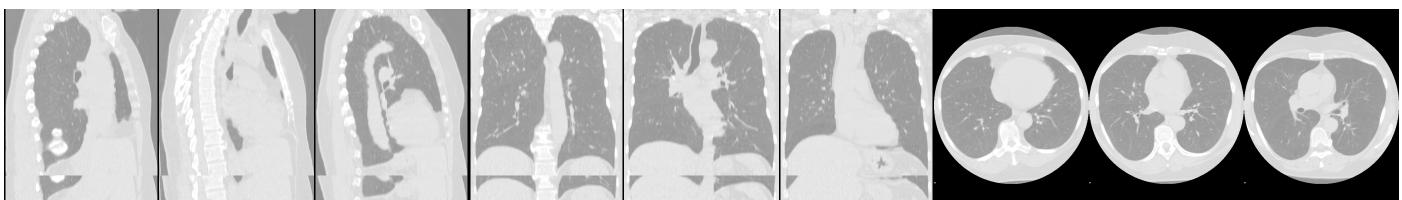
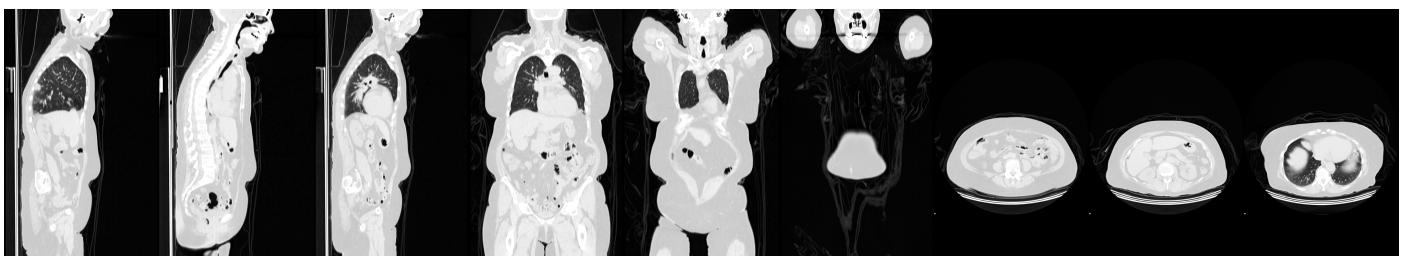


Figure 2: Case of Slice Distance check failure.



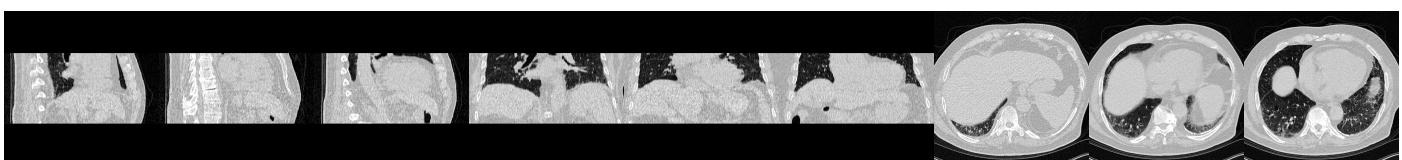
Figure 3: Case of Filtering too few slices.



Case of Physical Length Filtering showing extension outside of the region of interest.



Case of Orientation check failure.



Case found in `slicesdir` sanity check failure.

References

1. **Slice Location Attribute – DICOM Standard Browser** <https://dicom.innolitics.com/ciods/ct-image/image-plane/00201041>
2. **The first step for neuroimaging data analysis: DICOM to NIfTI conversion**
Xiangrui Li, Paul S. Morgan, John Ashburner, Jolinda Smith, Christopher Rorden
Journal of Neuroscience Methods (2016-05) <https://doi.org/f8jkpx>
DOI: [10.1016/j.jneumeth.2016.03.001](https://doi.org/10.1016/j.jneumeth.2016.03.001) · PMID: [26945974](#)