Cairo University

Faculty of Engineering

Cairo University

Faculty of Engineering

Department of Systems & Biomedical Engineering

Interactive GUI Tool for CT Image Parameter Visualization

May 5, 2025

Ahmad Kamal
Amgad Atef
Muhammad Sami Ahmad
Muhammad Sayed Abdelsalam
Ziyad Hossam

Supervised by:
Dr. Ahmad Ehab
T.A. Ola Sarhan

1 Abstract

This report outlines the development and implementation of an advanced Graphical User Interface (GUI) tool designed to enhance the understanding of Computed Tomography (CT) image processing parameters using clinical DICOM data. The tool enables users to load a folder of DICOM files, forming a 3D volume, and interactively manipulate parameters such as reconstruction filters (applied in frequency or spatial domains), slice thickness (via interpolation or averaging with Mean, Gaussian, or Median filters), and windowing settings (Window Level and Window Width). Additional features include slice navigation, metadata display, preset windowing options, and export functionality for processed slices. Developed using Python with PyQt5 for the GUI, Matplotlib for visualization, NumPy for numerical operations, pydicom for DICOM handling, and Scikit-image/SciPy for image processing, the application serves as a robust educational and analytical platform for exploring the impact of CT imaging parameters.

2 Introduction

Computed Tomography (CT) is a cornerstone of medical imaging, providing detailed cross-sectional views of patient anatomy. The appearance of CT images is influenced by acquisition parameters, reconstruction algorithms, and display settings, including reconstruction filters, slice thickness, and windowing adjustments. Understanding these parameters is critical for optimizing diagnostic image quality. This project aimed to create an interactive desktop application that allows users to explore these effects on real clinical data. Specifically, we utilized the "stageii-colorectal-ct" dataset from The Cancer Imaging Archive (TCIA) (`https://www.cancerimagingarchive.net/collection/stageii-colorectal-ct/`), which comprises abdominal or pelvic enhanced CT images from 230 patients with stage II colorectal cancer, providing a rich resource for testing and demonstration.

The tool's objectives are:

1. Load and process 3D CT volumes from DICOM folders.

2. Provide interactive controls to adjust and visualize:

- Reconstruction filter types (Ramp, Hann, Hamming, None) in frequency or spatial domains.
- Slice thickness via interpolation (thinner slices) or averaging (thicker slices) with Mean, Gaussian, or Median filters.
- Window Level (WL) and Window Width (WW) with preset options for Brain, Abdomen, Bone, and Spine.

3. Enable real-time image updates, slice navigation, metadata display, and export of processed slices.

3 Theoretical Background

This section describes the CT imaging parameters controlled by the GUI tool, focusing on their theoretical foundations and practical implications.

3.1 Reconstruction Filters

Reconstruction filters are integral to CT image reconstruction, particularly in Filtered Back-Projection (FBP). They modify the frequency content of projection data to improve image quality.

Purpose of Filters:

- *Blurring Correction:* Back-projection introduces blurring, which filters counteract.
- *Noise Control:* Filters balance noise amplification and image sharpness.

Filter Application:

1. *Frequency Domain:* Filters are applied to the Fourier Transform of the image data. The tool simulates this by applying Ramp, Hann, or Hamming filters to the 2D FFT of each slice.
2. *Spatial Domain:* Filters are applied via convolution with kernels (e.g., Laplacian for Ramp, window-based for Hann/Hamming), mimicking post-processing effects.

Filter Types:

- *Ramp:* A high-pass filter that maximizes sharpness but amplifies noise.
- *Hann/Hamming:* Smoother filters that reduce noise at the cost of some detail.
- *None:* No filtering, preserving the original image.

3.2 Slice Thickness

Slice thickness determines the longitudinal (z-axis) extent of tissue contributing to each CT slice.

Effects:

- *Thicker Slices:* Reduce noise via averaging but lower z-axis resolution, increasing partial volume effects.
- *Thinner Slices:* Improve resolution but increase noise.

Simulation in the Tool:

- *Thicker Slices:* Simulated by averaging multiple slices using Mean, Gaussian, or Median filters, reducing noise and detail.
- *Thinner Slices:* Achieved through interpolation (zooming) of the 3D volume, increasing slice count.

3.3 Windowing (Window Level / Window Width)

CT images span a wide range of Hounsfield Units (HU), but displays are limited to 256 gray levels. Windowing maps a subset of HU values to the grayscale range.

- *Window Width (WW):* Defines the HU range mapped to the grayscale spectrum, affecting contrast.
- *Window Level (WL):* Sets the center HU value, controlling brightness.

The tool allows manual WL/WW adjustments and provides presets.

Presets: The tool includes predefined WL/WW pairs optimized for specific tissues:

- *Brain:* WL=80, WW=40
- *Liver:* WL=150, WW=30
- *Bone:* WL=1800, WW=400
- *Soft Tissue:* WL=400, WW=40

3.4 Parameter Optimization

Choosing optimal reconstruction filters, slice thickness, and windowing settings involves trade-offs between resolution, noise, and contrast, tailored to the diagnostic task. The GUI facilitates exploration of these trade-offs through interactive controls and real-time visualization.

4 Implementation Details

4.0.1 Software Stack

The tool leverages the following technologies:

- **Python 3:** Core programming language.
- **PyQt5:** GUI framework for interactive controls and layout.
- **Matplotlib:** Visualization of processed images and navigation toolbar.
- **NumPy:** Array operations for image processing and windowing.
- **pydicom:** Reading and parsing DICOM files for 3D volume creation.
- **Scikit-image/SciPy:** Image processing, including filtering and interpolation.
- **imageio:** Exporting processed slices as images.

4.0.2 Architecture

The GUI is divided into two main areas:

- **Control Panel:**
  - *DICOM Folder Selection:* QPushButton to select a folder of .dcm files.
  - *Metadata Display:* QLabel showing Patient ID, slice count, resolution, and thickness.
  - *Reconstruction Filter Controls:* QComboBox for filter type (Ramp, Hann, Hamming, None) and QRadioButtons for domain (frequency/spatial).
  - *Slice Controls:* QSlider for slice navigation, QSpinBox for thickness (1–30 mm), QComboBox for averaging filter (Mean, Gaussian, Median).
  - *Windowing Controls:* QSliders for WL (-1000 to 3000 HU) and WW (1 to 2000 HU), QComboBox for presets.
  - *Reset Button:* QPushButton to restore default parameters.
- **Image Display Area:**
  - Matplotlib FigureCanvas for rendering the processed slice.
  - Navigation toolbar for zooming, panning, and saving.
  - Context menu (right-click) for exporting all slices as a tiled image.

4.0.3 Core Workflow

The application operates as follows:

1. *Initialization:* GUI is created with disabled controls until a DICOM folder is loaded.
2. *DICOM Loading:* User selects a folder; .dcm files are loaded into a 3D NumPy array, sorted by InstanceNumber, with metadata extracted.
3. *Parameter Adjustment:* Users adjust filter type, domain, slice thickness, averaging filter, and windowing via interactive controls.
4. *Processing:* Each parameter change triggers:
   - Slice thickness adjustment (interpolation for thinner slices, averaging for thicker slices).
   - Reconstruction filter application (FFT-based for frequency domain, convolution for spatial domain).
   - Windowing to map HU to grayscale.
5. *Concurrency:* Resource-intensive tasks (e.g., interpolation) run in separate threads to prevent GUI freezing, with UI elements disabled during processing.
6. *Real-time Update:* Processed slice is displayed, with progress dialogs for interpolation/averaging.
7. *Export:* Users can export all slices as a tiled PNG image via the context menu.

4.0.4 Export Functionality

Processed slices can be exported as a tiled PNG image via a right-click context menu:

- *Tiling Logic:* Slices are arranged in a grid using `matplotlib.subplots`, with unused subplots hidden.
- *Resolution:* Images are saved at 300 DPI using `imageio` for diagnostic-quality output.
- *Metadata Inclusion:* Each subplot is labeled with its slice index for traceability.

4.0.5 Error Handling and User Feedback

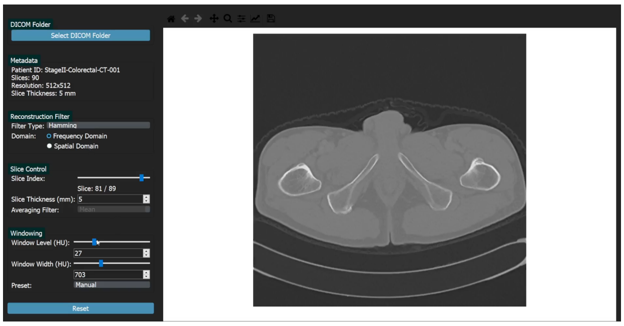
The tool implements robust error handling to guide users:

- *DICOM Loading Errors:* Invalid folders or non-DICOM files trigger `QMessageBox` alerts with descriptive messages (e.g., "No DICOM files found").
- *Progress Dialogs:* Long operations (e.g., interpolation) display `QProgressDialog` with cancellation support.
- *Input Validation:* Slice indices and thickness values are clamped to valid ranges to prevent out-of-bounds errors.

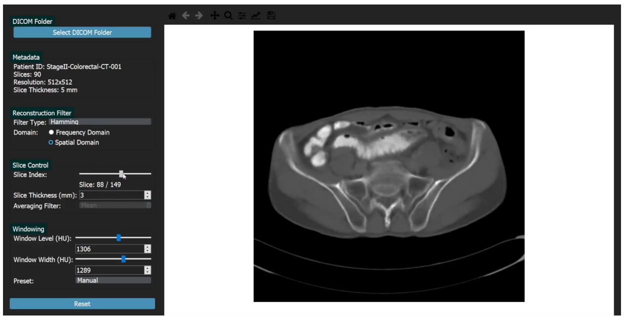
5 Results and Demonstration

The following figures demonstrate key functionalities of the CT Reconstruction Tool using the StageII-Colorectal-CT-001 dataset. Parameters correspond to settings shown in the GUI screenshots and reflect the tool's ability to modify reconstruction filters, slice thickness, and windowing.

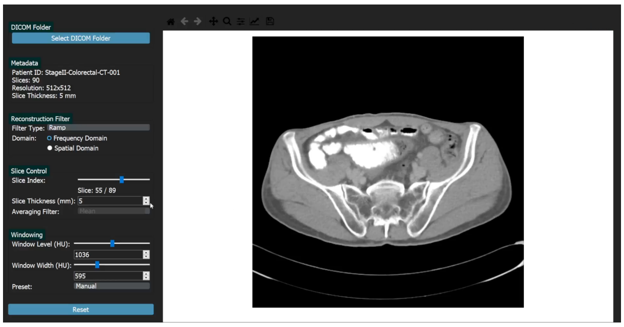
(a) Windowing with Ramp filter (spatial domain). Slice thickness: 5 mm. Window Level: 1360 HU, Window Width: 855 HU. The Ramp filter enhances edge sharpness at the cost of increased noise.



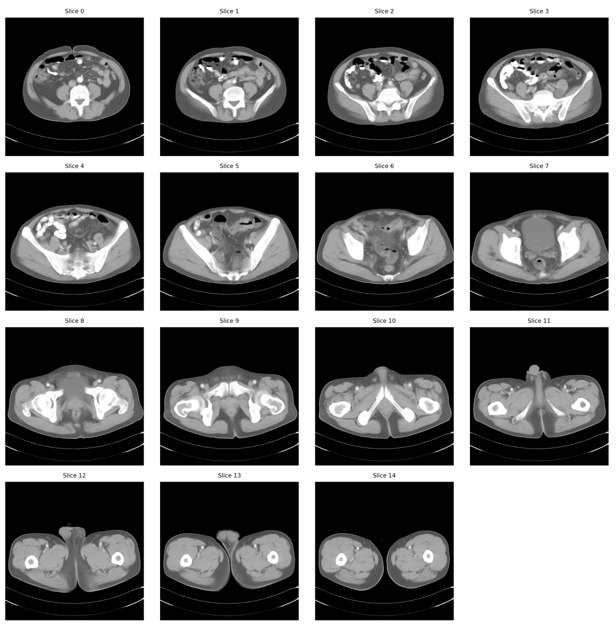
(b) Windowing with Hamming filter (spatial domain). Slice thickness: 5 mm. The Hamming filter balances noise reduction and detail preservation.



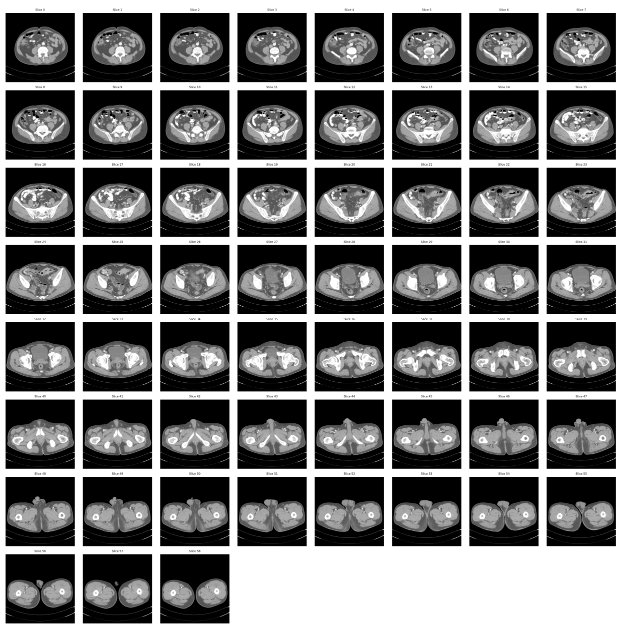
(c) Thin slices (3 mm) with Hamming filter (frequency domain). Window Level: 1306 HU, Window Width: 1289 HU. Thinner slices improve z-axis resolution but require noise-reducing filters.



(d) Default 5 mm slices with Ramp filter (spatial domain). Demonstrates baseline slice thickness and high-pass filtering for structural clarity.



(e) Thick slices (20 mm) using averaging. Reduces noise through slice aggregation but increases partial volume effects.



(f) Slice navigation interface showing 5 mm slices. Total 149 slices with interactive index control.

Figure 1: Visualization of CT parameter adjustments. (a-b) Windowing trade-offs between contrast and brightness. (c-d) Slice thickness and filter interactions. (e-f) Thickness extremes and navigation.

Key observations:

- Frequency-domain filters (Fig. 1c) require FFT operations, while spatial filters (Fig. 1a) use convolution.
- Thicker slices (Fig. 1e) exhibit smoother textures due to averaging, while thinner slices (Fig. 1c) retain finer details.
- Window Width adjustments (Fig. 1a vs Fig. 1b) directly impact tissue contrast.