electronics

MDPI

Article

DicomOS: A Preliminary Study on a Linux-Based Operating System Tailored for Medical Imaging and Enhanced Interoperability in Radiology Workflows

Tiziana Currieri [1,*], Orazio Gambino [2], Roberto Pirrone [2] and Salvatore Vitabile [1,*]

[1] Department of Biomedicine, Neuroscience and Advanced Diagnostics (BiND), University of Palermo, 90127 Palermo, Italy

[2] Department of Engineering, University of Palermo, 90128 Palermo, Italy; orazio.gambino@unipa.it (O.G.); roberto.pirrone@unipa.it (R.P.)

* Correspondence: tiziana.currieri@unipa.it (T.C.); salvatore.vitabile@unipa.it (S.V.)

Abstract: In this paper, we propose a Linux-based operating system, namely, DicomOS, tailored for medical imaging and enhanced interoperability, addressing user-friendly functionality and the main critical needs in radiology workflows. Traditional operating systems in clinical settings face limitations, such as fragmented software ecosystems and platform-specific restrictions, which disrupt collaborative workflows and hinder diagnostic efficiency. Built on Ubuntu 22.04 LTS, DicomOS integrates essential DICOM functionalities directly into the OS, providing a unified, cohesive platform for image visualization, annotation, and sharing. Methods include custom configurations and the development of graphical user interfaces (GUIs) and command-line tools, making them accessible to medical professionals and developers. Key applications such as ITK-SNAP and 3D Slicer are seamlessly integrated alongside specialized GUIs that enhance usability without requiring extensive technical expertise. As preliminary work, DicomOS demonstrates the potential to simplify medical imaging workflows, reduce cognitive load, and promote efficient data sharing across diverse clinical settings. However, further evaluations, including structured clinical tests and broader deployment with a distributable ISO image, must validate its effectiveness and scalability in real-world scenarios. The results indicate that DicomOS provides a versatile and adaptable solution, supporting radiologists in routine tasks while facilitating customization for advanced users. As an open-source platform, DicomOS has the potential to evolve alongside medical imaging needs, positioning it as a valuable resource for enhancing workflow integration and clinical collaboration.

Keywords: graphical user interfaces; customizable Linux distribution; medical imaging operating system; clinical diagnostics software; bash script

check for updates

Academic Editor: Pawel Strumillo

Received: 29 December 2024
Revised: 11 January 2025
Accepted: 14 January 2025
Published: 15 January 2025

Citation: Currieri, T.; Gambino, O.; Pirrone, R.; Vitabile, S. DicomOS: A Preliminary Study on a Linux-Based Operating System Tailored for Medical Imaging and Enhanced Interoperability in Radiology Workflows. *Electronics* **2025**, *14*, 330. https://doi.org/10.3390/electronics14020330

1. Introduction

Operating systems form the backbone of computational environments, managing resources, executing processes, and serving as the interface between hardware and software applications. In specialized fields such as medical imaging, where precision and reliability are critical, an operating system must support efficient task management and meet the rigorous demands of clinical diagnostics and research. Among the available platforms, Linux [1] stands out for its open-source nature, flexibility, and active developer community, making it a highly adaptable alternative to proprietary systems like Windows and macOS. This adaptability is exemplified by projects such as Debian Med [2], which integrates various medical tools within a Linux-based environment, primarily focusing on ease of installation and software compatibility. While Debian Med excels in providing a pre-configured ecosystem for various medical applications, it relies on third-party tools for imaging workflows, which require additional installation and configuration to support advanced functionalities such as DICOM integration. Similarly, MONAI [3], an open-source framework optimized for deep learning in healthcare, offers powerful tools for training AI models but requires significant technical expertise and does not address the broader challenges of workflow integration in clinical settings. In contrast, DicomOS combines ease of use, pre-configured tools for imaging, and advanced customization tailored to clinical workflows, bridging the gaps identified in these projects. Table 1 summarizes the main differences between DicomOS, Debian Med, and MONAI, highlighting the unique features of each and their suitability for specific use cases in medical imaging. Thanks to its open architecture, Linux enables unrestricted customization of core functionalities, allowing for the creation of environments specifically tailored to clinical needs. Several Linux-based distributions illustrate its adaptability in specialized fields. For example, Neurodebian [4] and Link4neuro [5] provide tools optimized for neuroscience and neuroimaging. Platforms like MITK [6] and OHIF [7] further demonstrate how open-source imaging software enhances accessibility and usability in targeted domains. Neurodebian integrates tools such as FSL [8], AFNI [9], and FreeSurfer [10], creating a unified ecosystem that simplifies setup and ensures software compatibility. These customized environments highlight Linux's capacity to support field-specific requirements while fostering reproducibility and collaboration, underscoring its potential as a foundation for dedicated radiological systems. The lack of interoperability between specialized software tools in radiology often disrupts workflows. Clinicians frequently rely on platform-specific applications to visualize, analyze, and share diagnostic images, creating barriers to seamless collaboration. For example, a radiologist using a Windows-based tool may encounter compatibility issues with a colleague working on macOS, resulting in delays and inefficiencies. The fragmented nature of medical imaging software exacerbates this problem, with some tools offering only basic visualization while others provide complex, feature-rich interfaces. Recent developments in data-driven graphical user interfaces for DICOM have proposed solutions to dynamically adapt functionalities based on the specific diagnostic scenario, addressing some of these usability challenges [11]. This forces radiologists to switch between applications to complete a single task, such as isolating regions of interest or applying image filters. Consequently, this fragmented approach increases the cognitive load and detracts from the primary objective of efficient diagnosis and reporting. To address these limitations, we present DicomOS, a Linux-based operating system tailored to the needs of radiologists and imaging specialists. DicomOS integrates core Digital Imaging and Communications in Medicine (DICOM) functionalities into a unified platform built on Ubuntu 22.04 LTS. Integrating DICOM functionality natively into the operating system presented several challenges, including ensuring compatibility with diverse DICOM implementations, maintaining performance for large imaging datasets, and creating user-friendly interfaces for accessing complex functionalities. These obstacles were addressed by leveraging widely used libraries such as PyDicom and developing automation scripts to simplify workflows, ensuring seamless integration into the operating system. By consolidating essential imaging operations such as annotation, visualization, and data sharing, DicomOS reduces the dependence on multiple software tools, streamlining workflows and lowering cognitive demand. Radiologists can perform key tasks within a cohesive environment, improving efficiency and collaboration. The open-source nature of Linux also allows DicomOS to be customized for institutional or individual needs, supporting emerging imaging techniques and advanced analytical tools as they become available. DicomOS offers both a graphical user interface and command-line tools, ensuring accessibility for clinicians who require straightforward applications and developers seeking robust customization options. By overcoming the fragmentation of current imaging systems and enabling interoperable, adaptable workflows, DicomOS is designed to enhance workflow integration and facilitate clinical collaboration by consolidating key medical imaging functionalities into a unified platform.

The main contributions of this research are highlighted as follows:

- DicomOS is designed as a general-purpose platform, making it adaptable for various specialities such as radiology, cardiology, and oncology.
- DicomOS provides a user-friendly interface for clinicians and a command-line environment for developers, allowing medical professionals to perform routine tasks while enabling programmers to customize workflows.
- Essential DICOM functions, visualization, annotation, and data manipulation, are built directly into the operating system, reducing the need for multiple external applications and simplifying workflows.
- Built on Linux, DicomOS allows for continuous improvement and adaptation to meet evolving imaging needs, offering flexibility that proprietary systems cannot match.

Table 1. Comparison of DicomOS with other open-source systems for medical imaging. The table highlights the key features of each system, focusing on clinical functionalities and customization capabilities.

Feature	DicomOS	Debian Med	MONAI
DICOM Support	Native integration with visualization, annotation, and data management capabilities	Limited support through third-party applications	No direct DICOM integration, focused on AI-based medical imaging workflows
Ease of Use	User-friendly graphical interface for clinicians and command-line tools for developers	Primarily oriented toward Linux experts	Requires advanced technical expertise for implementation and use
Integrated Tools	ITK-SNAP, 3D Slicer, and custom GUIs	Extensive ecosystem, but lacks automated integration	Powerful tools for AI training, but without GUI functionalities
Accessibility	Under development for ISO image distribution on physical hardware	Available as Debian packages	Open-source framework for Python
Customization	Highly customizable due to its open-source nature and Linux foundation	Limited to the compatibility of available packages	Extendable via Python modules, focusing on AI workflows rather than clinical needs

2. Development and Customization of DicomOS: Adapting Ubuntu for Medical Imaging

2.1. System Setup and Customization

To develop DicomOS, the base system selected was Ubuntu 22.04 Desktop LTS (Long-Term Support) [12], chosen for its robustness, comprehensive documentation, and compatibility with open-source medical imaging tools [13]. The installation was performed within a VirtualBox 6.1 [14] virtual environment, running on a macOS High Sierra desktop system with a 2.3 GHz Intel Core i5 processor and 8GB 1333 MHz DDR3 memory. This approach allowed for flexibility during the customization process, allowing for safe testing without interfering with existing hardware configurations. Once the base installation was completed, the VirtualBox Guest Additions were installed to enhance the compatibility between the virtual machine and the host system. These additions provided seamless integration features, such as improved graphical performance and shared clipboard functionality, ensuring a better user experience in the virtualized Ubuntu environment.

2.2. Customization of the Visual Environment

Following the installation, significant effort was devoted to customizing Ubuntu's visual elements to meet the needs of medical professionals while maintaining an intuitive and recognizable user interface. The Canta theme and Vimix icons were selected for their modern and visually appealing design, which aimed to create an inviting environment for users less familiar with Linux systems. The Vimix theme was chosen for its similarity to the macOS interface, a widely adopted medical imaging platform due to its user-friendly design [15]. This customization helped bridge the transition for radiologists accustomed to macOS. The customization process involved manually modifying the GNOME desktop environment by integrating these themes and additional extensions. One notable enhancement was installing the "Dash to Dock" GNOME extension, which replaced the default dock with a more accessible and customizable interface. This addition was designed to provide a familiar experience for users transitioning from other operating systems, ultimately improving the usability [16].

2.3. Branding and Identity Customization

Another key customization focused on creating a unified DicomOS brand identity, starting with the boot splash screen, the first visual element encountered by users during system startup. Plymouth [17], a graphical boot loader used in Linux systems, manages the animations and visuals displayed during the boot sequence. The default Ubuntu splash screen was replaced with a customized version featuring the DicomOS branding and a distinctive logo reflecting its focus on medical imaging. This process required modifications to Plymouth's themes and configuration files, allowing the default visuals to be replaced with a branded version unique to DicomOS. The login screen was updated to include DicomOS-specific branding and provide a consistent user experience. General user credentials were pre-configured to ensure easy initial access, while individual accounts could be created later to personalize the environment for different medical professionals. Figure 1 illustrates the customized DicomOS interface, showcasing its modern themes, icons, and user-centered design. This tailored visual setup highlights the system's adaptability and focus on usability.

Figure 1. Screenshot of the DicomOS interface, showing the customized theme, icons, and new graphical user interface applications tailored for clinical use.

2.4. Integration of Medical Applications and Usability Enhancements

One of the primary objectives of DicomOS was to incorporate essential medical imaging tools directly into the operating system. Popular applications such as ITK-SNAP [18] and 3D Slicer [19] were installed, as they provide advanced visualization and segmentation tools critical for radiological analysis [20,21]. However, one major limitation of these applications is that they do not natively support direct execution with a double click in Ubuntu. To make these applications more accessible to users with limited technical expertise, additional steps were taken to create executable versions. For 3D Slicer, the process began with modifying file permissions to allow for the installation of necessary extensions. Following this, a custom script was created to simplify the execution of 3D Slicer. This script was stored in `/usr/bin` to make it accessible from any terminal instance. Initially, the script was limited to a specific version of 3D Slicer, which posed a problem for future upgrades. A more generalized approach was implemented to address this by modifying the script to work with any version of 3D Slicer installed in a directory named with a pattern such as `Slicer*amd*`. This ensured compatibility with future versions, making the solution more flexible. The next step involved creating a desktop launcher to allow 3D Slicer to be executed with a double-click, making it accessible like any native application. The `.desktop` file was created to include the executable path, the icon path (using the official 3D Slicer logo for a professional appearance), and relevant category information to ensure it appeared in the correct sections of the system's application menu. This file was saved in the `/usr/share/applications` directory, integrating it into the system-wide application registry. This step significantly improved the usability, as medical professionals could launch 3D Slicer without interacting with the terminal [16]. For ITK-SNAP, a similar desktop launcher was configured to ensure consistency across the applications available in DicomOS. By leveraging the same principles used for 3D Slicer, ITK-SNAP was integrated into the system with minimal modifications, providing a seamless user experience. During the integration process, certain challenges were encountered, such as ensuring compatibility with multiple versions of the applications and maintaining stability across different setups. These challenges were addressed by iterative testing and optimization. Testing included verifying startup times, functionality across versions, and usability. Both 3D Slicer and ITK-SNAP were pre-existing and widely used applications. The modifications focused exclusively on enhancing accessibility by enabling double-click execution and ensuring proper integration into the DicomOS environment. Figure 2 illustrates the workflow to integrate them into DicomOS, highlighting key steps such as script creation and desktop launcher configuration. This figure will be explained in detail in the subsequent sections. Another enhancement focused on streamlining the day-to-day workflow for medical professionals and system administrators. To make common tasks such as updating the system, restarting, or shutting down more accessible, user-friendly command aliases like `upgrade`, `restart`, and `shutdown` were created. These aliases serve as shortcuts to frequently used commands, allowing users to execute these tasks more efficiently without remembering complex command syntax. Adding these aliases to the shell configuration files became immediately available for use. This enhancement reduced the cognitive load for users who may not be familiar with Linux terminal commands, thereby improving the efficiency and usability [16,22].

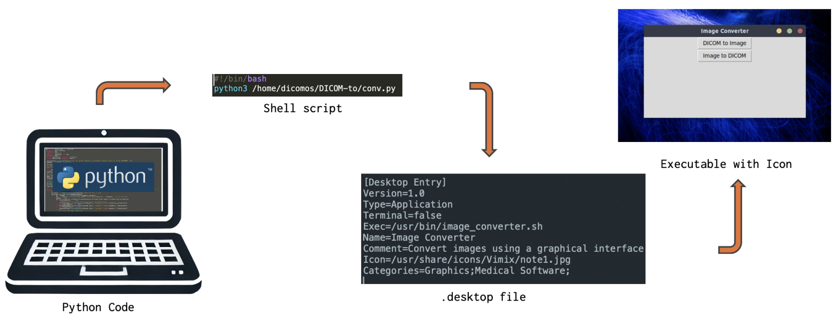

Figure 2. Example process for creating GUI executables in DicomOS, showing Python code execution through a shell script and desktop entry to facilitate easy user access.

3. Integration of Medical Workflows and Command-Line Tools

The primary objectives of DicomOS were to incorporate essential medical imaging tools directly into the operating system, providing an intuitive platform for medical professionals while supporting deeper customization options for developers and system administrators.

3.1. Development of Graphical User Interfaces for Medical Imaging

To enhance accessibility for different types of users, DicomOS is designed with two primary interaction pathways: one tailored for medical professionals who require straightforward graphical interfaces, and another optimized for programmers and system administrators who prefer command-line access for automation and customization. The usability improvements for medical professionals were developed based on established best practices for medical imaging workflows, and future evaluations will involve real-world testing with clinicians to refine these features further. Figure 3 illustrates the dual workflow in DicomOS, where Python scripts connect with shell commands and .desktop files to create accessible graphical user interface (GUI) applications. This setup enables clinicians to interact with advanced imaging tools through straightforward, executable interfaces without directly using Python scripts. These GUIs are designed to make tasks like image annotation, anonymization, and conversion accessible to users with limited technical expertise, leveraging clear visual menus and double-click execution to lower the technical barrier. Meanwhile, programmers have terminal access to the Python scripts, which they can execute using command-line commands and consult detailed manual pages (man pages) for customization and integration, described further below. This dual approach bridges the gap between usability and flexibility, enabling seamless collaboration between non-technical users and technical experts. Clinicians benefit from an intuitive interface that simplifies complex imaging tasks, while developers can customize and automate operations without interfering with the graphical workflows. To develop these GUIs, DicomOS utilizes Python 3.9 along with libraries such as `Tkinter` [23] for interface design, `matplotlib` [24] for image visualization, and `pydicom` [25] for handling Digital Imaging and Communications in Medicine (DICOM) files [26]. These GUIs are designed to provide medical professionals with intuitive tools that do not require extensive knowledge of Linux systems or programming.

As seen in Section 2.4, where executable versions were created for applications like ITK-SNAP and 3D Slicer to improve accessibility, a similar approach was followed for the custom GUIs developed specifically for DicomOS. This process involved creating shell scripts to invoke the underlying Python code, configuring .desktop files to assign application icons and enabling double-click execution. The GUIs were then systematically deployed to a standardized directory structure, with application files organized under /opt, a location commonly reserved for third-party software installations. This ensures clear separation from user-specific files and system-level components while maintaining compatibility with Linux filesystem standards. To facilitate intuitive access, the .desktop configuration was extended to link the GUI applications to system-wide menus, providing a seamless integration into the graphical environment of the operating system. The applications were further enhanced by associating them with high-resolution, scalable icons stored within the operating system's icon directories, ensuring visual consistency with native applications. The integration process also included setting appropriate permissions for executables and configuration files, ensuring accessibility for all system users without compromising security or functionality. This method was chosen after extensive testing with tools like PyInstaller [27], which, while designed for creating standalone executables from Python scripts, proved insufficient for this project. The GUIs in DicomOS depend on many external libraries and complex dependencies that PyInstaller could not fully resolve, resulting in incomplete or non-functional executables. By contrast, using shell scripts allowed for greater flexibility in managing these dependencies, ensuring that all required libraries could be properly loaded within the Python environment. Additionally, the .desktop configuration bridges the gap between complex software dependencies and user-friendly interfaces. By associating each GUI with a desktop entry, these applications appear alongside native system tools in the graphical environment, with consistent icons and support for double-click execution. This design abstracts technical details, allowing clinical users to interact with advanced imaging tools as if they were standard, pre-installed applications, eliminating the need for familiarity with Linux systems or command-line operations. This approach prioritizes practical implementation and adaptability, addressing the specific requirements of DicomOS without compromising its accessibility for clinical and technical users. Furthermore, it offers a flexible foundation that can be expanded or refined to accommodate future updates and additional features. Creating these executables is illustrated in Figure 2. The figure demonstrates how Python code is linked to an executable using a shell script and a .desktop entry to assign an icon and support double-click functionality. This makes the GUI applications accessible as native applications within the operating system. This approach effectively bridges the gap between the complexity of the development and the user experience, aligning with the broader objectives of DicomOS to improve usability and accessibility.

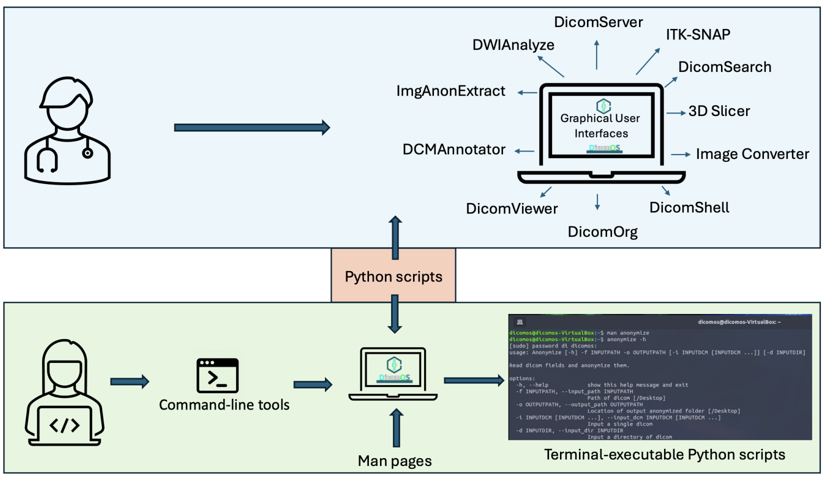

Figure 3. Workflow integration in DicomOS, demonstrating the development of GUI executables for medical professionals and command-line tools for programmers. The two sections are connected by a shared Python script layer, which supports GUI and command-line functionalities. This structure enables DicomOS to cater to the needs of both medical and technical users, providing an intuitive GUI for clinicians while offering direct, customizable access for programmers.

3.1.1. Dicom-Annotation GUI

One of the primary applications developed is **DCMAnnotator**, a program designed to facilitate radiologists' workflow and improve collaboration among specialists. This allows users to load image files in various formats, including DICOM, JPEG, and PNG and provides an interactive interface for the direct annotation of medical images. Users can add lines, rectangles, and textual notes to highlight areas of interest. Annotations are saved in a separate file associated with the original image, ensuring that the original data remain unaltered, which is crucial for maintaining data integrity.

The DCMAnnotator user interface is designed with simplicity and efficiency, adhering to best practices in human–computer interactions. The annotation tools are accessible via on-screen buttons and keyboard shortcuts, which enhance workflow efficiency. Specifically, the application provides buttons to draw lines and rectangles and to add prominently displayed text annotations for easy access. Keyboard shortcuts are implemented to expedite the annotation process; pressing `Ctrl+L` activates the line drawing tool, `Ctrl+R` activates the rectangle tool, and `Ctrl+T` activates the text annotation tool. This dual modality of interaction, which combines on-screen buttons with keyboard shortcuts, caters to users with different preferences and promotes a more efficient workflow. Keyboard shortcuts were integrated using event-handling mechanisms within the application. The program captures key press events and maps them to the corresponding annotation tools based on a predefined shortcut keymap. This feature allows users familiar with keyboard commands to switch tools rapidly without relying on mouse interactions, reducing the time required for annotations and minimizing interruptions during the diagnostic process. Implementing robust data management strategies addressed technical challenges such as ensuring reliable saving and loading annotations across sessions. Annotations are stored with detailed metadata, including object type, spatial coordinates, and textual content. The application saves these annotations in a separate file with a suffix indicating annotations, which allows for the consistent retrieval and display of annotations in future sessions without modifying the original image files.

3.1.2. Image Converter GUI

Another GUI application developed is **Image Converter**, which facilitates the conversion between DICOM files and common image formats such as JPEG and PNG. This application allows users to select a single DICOM file or a directory containing multiple DICOM files and convert them into image formats. Conversely, it can convert image files into DICOM format. The conversion process involves normalizing the pixel data and properly handling the image metadata. When converting images to DICOM, the application prompts the user to input essential metadata fields such as the patient name, study description, and series description. This ensures that the resulting DICOM files contain the necessary information for clinical use and compliance with DICOM standards. The user interface of Image Converter is designed to be straightforward, with buttons for selecting the conversion direction ("DICOM to Image" or "Image to DICOM") and dialogues for selecting input files or directories and specifying the output destination. Error handling mechanisms are implemented to inform the user of any issues during the conversion process, enhancing robustness and user experience. The application provides feedback upon successful conversion and includes the normalization of image pixel data to ensure consistent image quality.

3.1.3. DICOM Anonymizer GUI

In addition, the **ImgAnonExtract** was developed as a user-friendly tool for ensuring the privacy of DICOM files. The application allows users to select individual files or entire directories and specify which metadata fields to redact, such as patient ID, patient name, study date, institution name, and referring physician name. A set of checkboxes enables users to customize the redaction process, with the option to overwrite the original files or save processed copies to a designated location. A backup function is included to prevent data loss, allowing duplicates of the original files to be created before any modifications. ImgAnonExtract also provides advanced features for metadata extraction, enabling users to save selected information in a CSV file. The customizable selection of metadata fields facilitates data analysis and record-keeping. An intuitive interface displays results and operational logs, offering detailed feedback on the anonymization process and highlighting any errors encountered. The application is designed to align with GDPR principles by providing customizable metadata redaction options, logging anonymization actions for traceability, and supporting the exclusion of sensitive data during metadata export. Combined with an intuitive interface, these measures ensure that medical imaging data are handled securely and transparently, facilitating compliance with privacy regulations and institutional policies [28]. For example, the tool provides a backup function to prevent data loss during the anonymization process and logs all operations to create a complete audit trail. It also supports filtering metadata fields during CSV export, enabling users to exclude sensitive information while retaining data essential for research and analysis. By empowering users with these customizable and secure features, ImgAnonExtract serves as a practical solution for maintaining GDPR compliance in medical imaging workflows.

3.1.4. DICOM File Search GUI

To aid in efficiently locating relevant images within large datasets, the **DicomSearch** application was developed. This GUI allows users to search and preview DICOM files within a selected directory based on specific metadata fields. Users can specify search criteria such as PatientID, StudyDate, or Modality. The application scans the directory, reads the DICOM files' metadata, and displays a list of files matching the search criteria. The interface includes options to preview the DICOM files, showing essential metadata for quick reference. Users can also open and view individual DICOM images directly from the application, facilitating the immediate inspection of images of interest.

3.1.5. DICOM File Organizer GUI

Managing large volumes of DICOM files can be challenging without proper organization. The **DicomOrg** application streamlines the management of DICOM files by organizing them into subfolders based on selected metadata criteria. Users can choose a criterion such as PatientID, StudyDate, or Modality, and the application automatically sorts and moves the DICOM files into corresponding subfolders within the selected directory. This organizational tool helps maintain an orderly file structure, making navigating and managing medical imaging data easier.

3.1.6. DICOM Command Shell GUI

To provide a more interactive experience for users who prefer graphical interfaces over command-line tools, the **DicomShell** GUI was developed. This application allows users to execute various commands related to DICOM file management through a graphical interface. The available commands include the following:

- `list`: lists all DICOM files in the selected directory.
- `view`: displays extended information about a specified DICOM file.
- `analyze`: allows for the selection of a criterion for histogram display to analyze the distribution of metadata fields.

- `extract`: extracts advanced image features from a DICOM file.
- `extract_all`: extracts features from all DICOM files in the directory.
- `compare`: compares two DICOM files using the Structural Similarity Index (SSIM) and shows the difference map [29].
- `annotate`: adds annotations to a specified DICOM file.

The application includes a help option that provides descriptions of available commands and their usage, enhancing user support and accessibility. By integrating these functionalities into a GUI, the application makes it easier for users without command-line experience to perform complex tasks.

3.1.7. DICOM Server Navigator GUI

With the increasing need for remote access to medical imaging data, the **DicomServer** was developed to facilitate interaction with remote DICOM servers. This application enables users to navigate the contents of a DICOM server, browse directories and files, and download selected items seamlessly. Users can select files or folders from the server and preview images or download them to a local directory. The application dynamically retrieves server content, decodes file paths to ensure the proper handling of special characters and spaces, and organizes files and folders into a dropdown menu for intuitive navigation. To enhance usability, the interface provides buttons for navigation, image viewing, and downloading, complemented by a responsive logging area that displays feedback on operations and logs any errors encountered during network communication. When users select a DICOM file, the application fetches the data from the server, processes the pixel array, and applies appropriate windowing parameters to render a visually accurate grayscale preview. Window centre and width values, extracted from the DICOM metadata, are used to optimize visualization. If these values are unavailable, the application calculates them dynamically based on pixel intensity distribution. The DicomServer also supports downloading single files or entire folders, with recursive downloading for nested directories. The downloaded files are stored in a user-specified local directory, and subfolders are automatically created to mirror the server structure. For DICOM files, metadata such as Patient ID and acquisition parameters are extracted and displayed in the log, providing additional context for clinical use. Robust error handling ensures reliability, including timeouts for server requests and safeguards against invalid DICOM files. By integrating these advanced functionalities, the DicomServer GUI provides a user-friendly and efficient platform for remote medical imaging data management, ensuring secure and streamlined access to critical resources for clinical and research purposes.

3.1.8. DWI Longitudinal Analysis GUI

For clinicians involved in longitudinal studies, the **DWIAnalyze** application provides advanced tools for analyzing diffusion-weighted imaging (DWI) data across different time points [30]. This application facilitates the comparison of acute and chronic DWI scans for the same patient by automating critical processes such as image registration, difference computation, and report generation. Users can input directories containing acute and chronic DWI scans, with the application automatically matching patient images based on standardized directory structures. Preprocessing steps, including Gaussian smoothing and intensity normalization, are applied to ensure the consistency and comparability of the data. The workflow begins with the automated alignment of acute and chronic scans through image registration. If images differ in size, the application resamples them to ensure compatibility. Once aligned, different images are computed, highlighting structural changes between the two scans. The application calculates statistical metrics to quantify these changes, including the mean difference, standard deviation, and maximum values.

In addition, it calculates the Structural Similarity Index (SSIM), a widely used metric to assess the degree of structural similarity between images. These quantitative analyses are complemented by a visual representation of the central slice of the difference map, color-coded for clarity. The PDF report generated provides a detailed summary of the findings. It includes the file path of the difference image, the computed metrics (mean, standard deviation, and maximum differences), and the SSIM score. The report also offers an interpretation of the results based on these metrics. For instance, high mean difference values may indicate significant structural alterations, while low variability (standard deviation) suggests uniform changes across the brain. The SSIM score is particularly informative, with lower values pointing to substantial structural changes and higher scores indicating stability between scans. The tool's reliability is further supported by its integration of validated preprocessing techniques, such as Gaussian smoothing and intensity normalization, which minimize artefacts and enhance data consistency. By leveraging established metrics like SSIM and generating comprehensive PDF reports, DWIAnalyze ensures that its outputs accurately reflect structural changes between scans, providing clinicians with robust, interpretable insights into patient progression or recovery. This approach aids clinicians in evaluating disease progression or recovery. For example, in a longitudinal study of a traumatic brain injury patient, the report may reveal regions with significant differences that warrant further investigation or confirm minimal changes indicative of recovery. By combining automated analysis, robust metrics, and visual outputs, DWIAnalyze streamlines longitudinal imaging studies, offering clinicians a powerful tool for informed decision-making and enhanced patient care.

3.1.9. Medical Image Editor GUI

The **DicomViewer** is an advanced tool that allows users to perform various image processing operations on medical images. It supports loading images in multiple formats, including DICOM, JPEG, and PNG. The application provides a range of functionalities, such as the following:

- Image enhancement operations like adjusting contrast, brightness, and saturation.
- Applying filters such as sharpening, smoothing, Gaussian blurring, and histogram equalization.
- Performing geometric transformations like flipping images horizontally or vertically.
- Edge detection using Sobel and Canny operators.
- Noise addition and image denoising techniques.
- Image cropping and resizing.

The interface includes menus and toolbar icons for easy access to these functions, with visual feedback provided through real-time updates of the displayed image. The application supports batch processing, allowing users to apply operations to all images within a selected folder. Users can save the modified images and maintain a history of modifications to undo changes or restore the original image. Technical considerations in developing the Medical Image Editor included handling different image formats, managing color spaces, and ensuring efficient processing for large images. The application leverages OpenCV for image processing operations and provides an intuitive interface for non-technical users. These GUI applications were designed to be accessible to medical professionals without requiring programming skills or familiarity with command-line interfaces. By providing intuitive interfaces and guiding users through processes such as conversion, anonymization, searching, organizing, and editing, these tools aim to streamline workflows and reduce the potential for errors. The applications can be launched directly by double-clicking their icons, making them readily available within the DicomOS environment.

3.2. Development of Command-Line Tools for Programmers

Recognizing the needs of programmers and system administrators, DicomOS includes a suite of command-line tools as part of the **Dicom-to** package. These tools enhance the operating system's versatility by enabling the automation and customization of medical imaging tasks. The primary commands developed are `Convert`, `Extract`, `ConvertExtract`, and `Anonymize`.

The `Convert` command facilitates the conversion of DICOM files into common image formats such as JPEG, PNG, or BMP. This functionality is essential for integrating medical images into various workflows and for compatibility with software that may not natively support the DICOM format. Users can specify input and output paths, choose the desired image format, and apply the conversion to individual files or entire directories, providing flexibility in handling different volumes of data. Including a `-help` option provides users with detailed usage instructions directly from the terminal. The `Extract` command enables the extraction of metadata from DICOM files. Metadata extraction is crucial for analyzing imaging parameters, patient information, and other clinical data embedded within DICOM files. The extracted metadata is saved in a CSV file, which can be easily imported into data analysis tools or databases for further processing. Users can customize the metadata fields to be extracted by editing a configuration file in the `~/DICOM_to` directory. The `-help` option guides users on utilizing the command effectively. The `ConvertExtract` command combines the functionalities of both `Convert` and `Extract`, allowing for simultaneous image conversion and metadata extraction. This dual functionality streamlines workflows requiring image processing and data analysis, reducing the need for separate processing steps and enhancing efficiency. The `Anonymize` command facilitates the anonymization of DICOM files by removing sensitive patient information from metadata fields such as the patient name, patient ID, study dates, and institution information. Users can process individual files or entire directories and specify output paths for the anonymized files. Including the `-help` option ensures users have access to detailed instructions on command usage. These command-line tools were implemented using Python scripts that leverage libraries such as `pydicom` for DICOM file handling and `Pillow` for image processing. The scripts accept command-line arguments, enabling users to customize input paths, output destinations, and processing options. Error handling mechanisms are incorporated to manage invalid inputs and ensure robustness. To facilitate ease of use and adhere to Unix conventions [31], comprehensive man pages were created for each command. These man pages provide detailed documentation, including the command's purpose, available options, usage examples, and information on authorship and licensing. By installing the man pages in the system directory `/usr/local/share/man/man1`, users can access the documentation directly from the terminal using the `man` command. This approach supports effective integration into users' workflows and promotes consistency with standard Unix practices. For instance, the man page for the `Convert` command includes usage instructions such as the following:

```
Convert -f /Desktop -o /Documents -option jpg -i Img_x.dcm
```

This command converts the DICOM image `Img_xdcm` located on the desktop into a JPEG image, saving the output in the specified document directory within a `Final` folder. Creating the `Final` folder and subdirectories based on the execution date, chosen extension, and patient ID organizes the output systematically. Similarly, the man page for the `Extract` command provides examples like the following:

```
Extract -f /Documents -o /Desktop -d Directory
```

This command extracts metadata from all DICOM files within the directory named `Directory`, located in `/Documents`, and saves them in a CSV file called `Patient_Detail` on the desktop. Users can customize the metadata fields to be extracted by editing the `dicom_image_description.csv` file located in the `~/DICOM_to` directory.

The `ConvertExtract` command enhances productivity by combining these functionalities, as shown in its usage:

```
ConvertExtract -f /Desktop -o /Documents -option png -d Directory
```

This command converts DICOM images inside the `Directory` on the desktop into PNG images, placing them in the `Final` folder in `/Documents`, and simultaneously extracts metadata into `Patient_Detail.csv`. In addition to these commands, man pages were created for tools such as the `Anonymize` command, which facilitates the anonymization of DICOM files by removing sensitive patient information. The man page for `Anonymize` provides detailed instructions and examples, ensuring users can effectively utilize the tool while maintaining compliance with privacy regulations. Creating these man pages is a critical aspect of the tool development, as it ensures that users have immediate access to comprehensive documentation directly from the terminal. The man pages were carefully crafted to include sections such as Name, Synopsis, Description, Options, Examples, Copyright, and Authors, providing a thorough guide for each command. This practice aligns with Unix traditions and enhances the usability of the tools for programmers and system administrators. Comprehensive documentation, via man pages, equips users with readily accessible information on tool usage, options, and expected behaviours. This enhances usability and promotes consistency with standard practices in Unix-like operating systems, facilitating seamless integration into existing workflows. Developing these command-line tools and accompanying man pages demonstrates a commitment to supporting novice and experienced users. Programmers can incorporate these tools into scripts to automate repetitive tasks, such as batch conversion of images or systematic metadata extraction from large datasets. System administrators can utilize them to manage medical imaging repositories, ensure compliance with data standards, and facilitate data sharing while adhering to privacy regulations.

4. Results

The development and customization of DicomOS yielded a specialized Linux-based operating system tailored to the needs of medical imaging professionals. The primary outcome was successfully integrating essential medical imaging applications into a user-friendly environment, enhancing usability and workflow efficiency. By customizing the Ubuntu desktop environment with themes and icons resembling macOS, DicomOS provided an interface to reduce the learning curve for users transitioning from macOS. This visual consistency minimizes the learning curve of transitioning to a Linux-based operating system, enhancing user satisfaction and adoption rates [15]. Critical medical imaging applications like ITK-SNAP and 3D Slicer were seamlessly integrated into DicomOS. Custom execution scripts and desktop launchers were developed, enabling these applications to be executed directly from the desktop environment without needing command-line interaction. This integration streamlined workflows by providing quick access to advanced imaging tools, improving the efficiency of medical image analysis. Furthermore, a specialized GUI application suite was developed to address specific needs in medical imaging tasks. Dicom-Annotation, ConvertGUI, and DICOMAnonymizerGUI provided user-friendly interfaces for image annotation, conversion between DICOM and common image formats, and the anonymization of sensitive patient information, respectively. Each GUI tool, as summarized in Table 2, provides intuitive access to essential imaging functions, improving the overall user experience of DicomOS. These applications were designed following best practices in human–computer interaction, emphasizing simplicity and efficiency, including intuitive menus and keyboard shortcuts that facilitated a more efficient workflow, reducing the time required for routine tasks and minimizing potential errors. For programmers and system administrators, command-line tools such as `Convert`, `Extract`, and `Anonymize` were developed. These tools enhanced the versatility of DicomOS by enabling the automation and customization of medical imaging tasks, supporting large-scale data processing and integration into existing scripts and workflows. Comprehensive documentation was provided through adhering to Unix conventions, facilitating ease of use and consistency with standard operating procedures. In addition, integrating GUI and command-line tools allows DicomOS to meet different needs: clinicians can use intuitive GUIs, while developers and administrators can automate complex processes via command-line tools. This dual approach makes DicomOS versatile and suitable for various professional contexts. The project resulted in an operating system that effectively bridges the gap between complex medical imaging software and medical professionals' usability needs. By integrating essential tools and enhancing accessibility, DicomOS promotes efficiency and effectiveness in medical imaging workflows.

Table 2. Summary of GUIs integrated into DicomOS for medical imaging tasks.

GUI Application	Function
ITK-SNAP	Advanced medical image segmentation and visualization tool, often used for annotating structures in 3D medical images.
3D Slicer	Provides powerful tools for visualization and analysis, including segmentation, registration, and quantitative imaging.
DCMAnnotator	Allows users to annotate medical images with lines, rectangles, and text, saving annotations separately to preserve original image data.
Image Converter	Converts DICOM files to JPEG, PNG, and other formats, and vice versa, supporting metadata input for DICOM conversions.
ImgAnonExtract	Anonymizes selected metadata fields in DICOM files, with options to overwrite or save anonymized copies and create backups.
DicomSearch	Enables the search and preview of DICOM files based on metadata fields, facilitating the quick location of relevant images.
DicomOrg	Organizes DICOM files into subfolders by criteria such as PatientID or StudyDate, streamlining data management.
DicomShell	Provides a GUI for command-line operations like listing, viewing, analyzing, extracting features, and comparing DICOM files.
DicomServer	Allows browsing and downloading files from a remote DICOM server, including the preview and secure transfer of selected files.
DWIAnalyze	Compares acute and chronic DWI scans for a patient, performing image registration, difference computation, and report generation.
DicomViewer	Offers a range of image processing functions (e.g., contrast adjustment, filtering, and edge detection) for various medical image formats.

5. Discussion and Conclusions

5.1. Limitations

While the customization and development of DicomOS demonstrate the potential of adapting open-source operating systems for medical imaging, several challenges and limitations were encountered. A significant limitation of the current implementation is the absence of a distributable ISO (International Organization for Standardization) image for

DicomOS. Currently, DicomOS is distributed as a VirtualBox disk image file (`DicomOS.vdi`), enabling rapid deployment in virtualized environments. Although practical for testing and iterative development, this approach restricts accessibility to users unfamiliar with virtualization software or those requiring direct installation on physical hardware. Developing a universally deployable ISO image is a priority to address this limitation, ensuring broader usability and scalability across different infrastructures. To ensure a seamless transition to physical hardware, future development will focus on rigorous compatibility testing across various hardware configurations. This includes validating DicomOS on standardized components commonly found in medical imaging systems, such as GPUs for artificial intelligence (AI) workloads and high-performance storage solutions. The ISO image will include pre-configured drivers and automated setup scripts to simplify installation and minimize compatibility issues. Performance testing will also be conducted to compare the efficiency of DicomOS in virtualized versus physical environments, allowing the team to identify and resolve any discrepancies. These steps will ensure that DicomOS performs reliably and consistently across diverse hardware platforms. Another limitation involves compatibility with other Linux distributions. Although DicomOS is built on Ubuntu 22.04 LTS, its specific customizations may limit direct portability to distributions such as Fedora [32] or Arch Linux [33] without significant modifications. To address this, future iterations of DicomOS could leverage containerization technologies, such as Docker [34], to provide a distribution-independent runtime environment. Alternatively, packaging DicomOS as .deb and .rpm files or using configuration management tools like Ansible [35] could facilitate deployment across multiple Linux distributions. Additionally, scalability challenges may arise in larger radiology departments where simultaneous multi-user operations and higher volumes of imaging data could necessitate enhanced resource management. Scenarios involving much larger datasets or sophisticated imaging modalities, such as 3D reconstruction or high-resolution imaging, may further strain computational resources and require workflow optimization. To address these challenges, future iterations of DicomOS will explore the modular integration of GPU acceleration and support for parallel computing frameworks (e.g., CUDA or OpenCL) to enhance performance for computationally intensive tasks. These challenges could be mitigated by implementing a client–server architecture, optimizing the system for cloud-based deployment, or incorporating load-balancing solutions to ensure efficient processing and data access in high-demand environments. While DicomOS is highly customizable, adapting it to workflows involving emerging modalities may require additional development efforts, such as creating custom plugins or scripts tailored to specific imaging requirements. Comprehensive documentation and a growing community of users will support these adaptations, maintaining flexibility and scalability for diverse clinical and research scenarios. Furthermore, ensuring compatibility between the customized system components and updates to the underlying Ubuntu operating system requires ongoing maintenance. Additionally, the development of GUI applications introduces software maintenance and support complexity. Providing comprehensive documentation and user support is essential to help users fully leverage DicomOS's capabilities and address potential issues.

5.2. Future Directions

Future work will address the challenges revealed by the preceding analysis and explore opportunities for improvement. In particular, the next phase will involve conducting in-depth evaluations with a representative sample of users to validate the effectiveness of DicomOS in real-world clinical settings. Detailed data will be collected through structured tests with medical professionals and IT administrators, including metrics, such as task completion time, error rates, and user satisfaction scores. These analyses will provide essential feedback to drive iterative improvements, ensuring that the system continues to meet the evolving needs of its users. Additionally, future development will include comprehensive performance evaluations comparing DicomOS to other medical imaging platforms. These evaluations will measure processor speed, memory usage, and image rendering times to ensure competitive performance while maintaining its focus on usability and modularity. Moreover, expanding the range of integrated applications and tools, including support for emerging imaging modalities and advanced data analysis techniques [36], can enhance the utility and relevance of DicomOS in medical imaging. In particular, integrating AI-based tools represents a critical direction for future development. Recent studies, such as [37], illustrate how machine learning can improve the accuracy and efficiency of diagnostic workflows in radiology. In our work, planned AI tools include automatic segmentation, diagnostic prediction, and anomaly detection, all designed to improve workflow efficiency and diagnostic accuracy. These functionalities will leverage advanced frameworks such as TensorFlow and PyTorch, ensuring compatibility with existing clinical workflows and allowing for future expansions. Automatic segmentation modules offer clinicians an intuitive interface for processing DICOM images, applying segmentation algorithms, and exporting results in formats suitable for clinical reporting. Diagnostic prediction tools will generate interpretable outputs, such as probability scores and visual overlays, highlighting regions of interest to assist clinicians in decision-making. Additionally, anomaly detection algorithms could identify unexpected patterns in imaging data, providing early warnings for potential clinical issues. A potential enhancement for DicomOS includes integrating a GUI for melanoma classification using pre-trained weights from models like the Vision Transformer (ViT) described in [38]. This interface would allow clinicians to classify dermoscopic images by leveraging AI capabilities directly within the DicomOS environment, providing a streamlined and user-friendly workflow for diagnostic support. DicomOS will integrate these AI-based enhancements into its graphical environment to ensure practical usability. These tools will leverage the existing modular architecture of DicomOS, which supports the addition of Python-based plugins and .desktop executables. This architecture enables AI functionalities to be embedded as standalone modules, ensuring compatibility with the current GUI while maintaining consistent workflows for clinicians. This integration will prioritise user-friendliness, minimizing the learning curve for clinicians while maintaining advanced customization options for developers. Furthermore, including interactive tutorials and example workflows will facilitate the adoption and encourage the widespread use of these AI-powered features in diverse clinical settings.

5.3. Conclusions

In conclusion, the development of DicomOS showcases the potential for creating specialized operating systems that directly address the needs of medical imaging professionals. By integrating essential tools, enhancing usability, and providing both GUI and command-line options, DicomOS stands as a valuable resource that can improve efficiency and effectiveness in medical imaging workflows. Addressing current limitations, such as creating a distributable ISO image, and engaging with the user community are recommended to refine further and expand its capabilities, ensuring that it remains a relevant and useful tool in the rapidly evolving field of medical imaging. DicomOS exemplifies how open-source platforms can address specialized needs in professional domains, contributing to the discourse on customizable and cost-effective technological solutions in healthcare, and further positioning it as a model for innovation in clinical and research settings. By addressing current limitations and incorporating cutting-edge innovations, DicomOS has the potential to become an indispensable tool for enhancing efficiency, collaboration, and innovation in medical imaging workflows.

Author Contributions: Conceptualization, T.C. and O.G.; methodology, T.C.; software, T.C.; validation, T.C., O.G. and R.P.; formal analysis, T.C. and O.G.; investigation, T.C., O.G. and R.P.; resources, S.V.; writing—original draft preparation, T.C.; writing—review and editing, O.G., R.P. and S.V.; supervision, O.G., R.P. and S.V.; funding acquisition, S.V. All authors have read and agreed to the published version of the manuscript.

Funding: This contribution is partially supported by the Ministry of University and Research (MUR) as part of the FSE REACT-EU—PON 2014-2020 "Research and Innovation" resources—Green/Innovation Action—DM MUR 1061/2021.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial intelligence
DICOM	Digital Imaging and Communications in Medicine
GNOME	GNU Network Object Model Environment
GPU	Graphics Processing Unit
GUI	Graphical user interface
ISO	International Organization for Standardization
SSIM	Structural Similarity Index Measure

References

1. Torvalds, L. *Just for Fun: The Story of an Accidental Revolutionary*; Harper Audio: New York, NY, USA, 2001.
2. Möller, S.; Krabbenhöft, H.N.; Tille, A.; Paleino, D.; Williams, A.; Wolstencroft, K.; Goble, C.; Holland, R.; Belhachemi, D.; Plessy, C. Community-driven computational biology with Debian Linux. *BMC Bioinform.* **2010**, *11*, S5. [CrossRef]
3. Cardoso, M.J.; Li, W.; Brown, R.; Ma, N.; Kerfoot, E.; Wang, Y.; Murrey, B.; Myronenko, A.; Zhao, C.; Yang, D.; et al. Monai: An open-source framework for deep learning in healthcare. *arXiv* **2022**, arXiv:2211.02701.
4. Halchenko, Y.O.; Hanke, M. Open is not enough. Let's take the next step: An integrated, community-driven computing platform for neuroscience. *Front. Neuroinform.* **2012**, *6*, 22. [CrossRef]
5. Nemoto, K.; Dan, I.; Rorden, C.; Ohnishi, T.; Tsuzuki, D.; Okamoto, M.; Yamashita, F.; Asada, T. Lin4Neuro: A customized Linux distribution ready for neuroimaging analysis. *BMC Med. Imaging* **2011**, *11*, 3. [CrossRef]
6. Wolf, I.; Vetter, M.; Wegner, I.; Nolden, M.; Bottger, T.; Hastenteufel, M.; Schobinger, M.; Kunert, T.; Meinzer, H.P. The medical imaging interaction toolkit (MITK): A toolkit facilitating the creation of interactive software by extending VTK and ITK. In Proceedings of the Medical imaging 2004: Visualization, Image-Guided Procedures, and Display, SPIE, San Diego, CA, USA, 14–19 February 2004; Volume 5367, pp. 16–27.
7. Ziegler, E.; Urban, T.; Brown, D.; Petts, J.; Pieper, S.D.; Lewis, R.; Hafey, C.; Harris, G.J. Open health imaging foundation viewer: An extensible open-source framework for building web-based imaging applications to support cancer research. *JCO Clin. Cancer Inform.* **2020**, *4*, 336–345. [CrossRef]
8. Jenkinson, M.; Beckmann, C.F.; Behrens, T.E.; Woolrich, M.W.; Smith, S.M. Fsl. *Neuroimage* **2012**, *62*, 782–790. [CrossRef]
9. Cox, R.W. AFNI: What a long strange trip it's been. *Neuroimage* **2012**, *62*, 743–747. [CrossRef]
10. Fischl, B. FreeSurfer. *Neuroimage* **2012**, *62*, 774–781. [CrossRef]
11. Gambino, O.; Rundo, L.; Cannella, V.; Vitabile, S.; Pirrone, R. A framework for data-driven adaptive GUI generation based on DICOM. *J. Biomed. Inform.* **2018**, *88*, 37–52. [CrossRef]
12. Ubuntu: The Official Ubuntu Documentation. 2023 Available online: https://help.ubuntu.com/ (accessed on 10 October 2023).
13. Topol, E.J. High-performance medicine: The convergence of human and artificial intelligence. *Nat. Med.* **2019**, *25*, 44–56. [CrossRef]
14. Oracle Corporation. Oracle VM VirtualBox Version 6.1. 2023. Available online: https://www.virtualbox.org/ (accessed on 10 October 2023).
15. Ventola, C.L. Mobile devices and apps for health care professionals: Uses and benefits. *Pharm. Ther.* **2014**, *39*, 356.

16. Johnson, C.M.; Johnson, T.R.; Zhang, J. A user-centered framework for redesigning health care interfaces. *J. Biomed. Inform.* **2005**, *38*, 75–87. [CrossRef]
17. Plymouth Theming Guide. 2023 Available online: https://www.freedesktop.org/wiki/Software/Plymouth/ (accessed on 10 October 2023).
18. Yushkevich, P.A.; Gao, Y.; Gerig, G. ITK-SNAP: An interactive tool for semi-automatic segmentation of multi-modality biomedical images. In Proceedings of the 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Orlando, FL, USA, 6–20 August 2016; pp. 3342–3345.
19. Fedorov, A.; Beichel, R.; Kalpathy-Cramer, J.; Finet, J.; Fillion-Robin, J.C.; Pujol, S.; Bauer, C.; Jennings, D.; Fennessy, F.; Sonka, M.; et al. 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magn. Reson. Imaging* **2012**, *30*, 1323–1341. [CrossRef]
20. Kikinis, R.; Pieper, S.D.; Vosburgh, K.G. 3D Slicer: A Platform for Subject-Specific Image Analysis, Visualization, and Clinical Support. In *Intraoperative Imaging and Image-Guided Therapy*; Springer: New York, NY, USA, 2014.
21. Yushkevich, P.A.; Pashchinskiy, A.; Oguz, I.; Mohan, S.; Schmitt, J.E.; Stein, J.M.; Zukić, D.; Vicory, J.; McCormick, M.; Yushkevich, N.; et al. User-guided segmentation of multi-modality medical imaging datasets with ITK-SNAP. *Neuroinformatics* **2019**, *17*, 83–102. [CrossRef]
22. Kushniruk, A.W.; Patel, V.L. Cognitive and usability engineering methods for the evaluation of clinical information systems. *J. Biomed. Inform.* **2004**, *37*, 56–76. [CrossRef] [PubMed]
23. Lundh, F. An Introduction to Tkinter. 1999, Voulme 539, p. 540. Available online: www.pythonware.com/library/tkinter/introduction/index.htm (accessed on 21 April 2024).
24. Bisong, E.; Bisong, E. Matplotlib and seaborn. *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*; Apress: Berkeley, CA, USA, 2019; pp. 151–165.
25. Pydicom Community. pydicom: DICOM handling in Python. Available online: https://pydicom.github.io (accessed on 21 February 2023).
26. Pianykh, O.S. *Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 10.
27. PyInstaller Development Team. PyInstaller: Package Python Applications. Version 5.13. 2024. Available online: https://pyinstaller.org/en/stable/ (accessed on 6 December 2024).
28. Wilkinson, M.D.; Dumontier, M.; Aalbersberg, I.J.; Appleton, G.; Axton, M.; Baak, A.; Blomberg, N.; Boiten, J.W.; da Silva Santos, L.B.; Bourne, P.E.; et al. The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **2016**, *3*, 1–9. [CrossRef]
29. Brunet, D.; Vrscay, E.R.; Wang, Z. On the mathematical properties of the structural similarity index. *IEEE Trans. Image Process.* **2011**, *21*, 1488–1499. [CrossRef] [PubMed]
30. Bammer, R. Basic principles of diffusion-weighted imaging. *Eur. J. Radiol.* **2003**, *45*, 169–184. [CrossRef] [PubMed]
31. Kerrisk, M. *The Linux Programming Interface: A Linux and UNIX System Programming Handbook*; No Starch Press: San Francisco, CA, USA, 2010.
32. The Fedora Project. Fedora: The Fedora Project. Available online: https://getfedora.org (accessed on 15 January 2023).
33. Arch Linux Contributors. Arch Linux: A Lightweight and Flexible Linux Distribution. Available online: https://archlinux.org (accessed on 15 January 2023).
34. Docker Inc. Docker: Empowering App Development for Scalability. Available online: https://www.docker.com (accessed on 15 January 2023).
35. Red Hat Inc. Ansible: Simple Automation Across IT Environments. Available online: https://www.ansible.com (accessed on 24 July 2024).
36. Esteva, A.; Robicquet, A.; Ramsundar, B.; Kuleshov, V.; DePristo, M.A.; Chou, K.; Cui, C.; Corrado, G.S.; Thrun, S.; Dean, J. A guide to deep learning in healthcare. *Nat. Med.* **2019**, *25*, 24–29. [CrossRef]
37. Onakpojeruo, E.P.; Mustapha, M.T.; Ozsahin, D.U.; Ozsahin, I. Enhanced MRI-based brain tumour classification with a novel Pix2pix generative adversarial network augmentation framework. *Brain Commun.* **2024**, *6*, fcae372. [CrossRef]
38. Cirrincione, G.; Cannata, S.; Cicceri, G.; Prinzi, F.; Currieri, T.; Lovino, M.; Militello, C.; Pasero, E.; Vitabile, S. Transformer-based approach to melanoma detection. *Sensors* **2023**, *23*, 5677. [CrossRef] [PubMed]