A Web-based Dicom Image and Plane Viewer

Priya Darshini B
Department of Biomedical Engineering
KPR Institute of Engineering and Technology
Coimbatore, Tamil Nadu, India
priyabalu1708@gmail.com

Deepan Chakkaravarthy N
Department of Biomedical Engineering
KPR Institute of Engineering and Technology
Coimbatore, Tamil Nadu, India
chakkaravarthydc@gmail.com

Gokul B
Department of Biomedical Engineering
KPR Institute of Engineering and Technology
Coimbatore, Tamil Nadu, India
sivagokul07@gmail.com

Sabari Maharaja B
Department of Biomedical Engineering
KPR Institute of Engineering and Technology
Coimbatore, Tamil Nadu, India
sabari03102001maharaja@gmail.com

2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT) | 978-1-6654-9360-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICECCT56650.2023.10179664

***Abstract* - Preclinical research, clinical diagnosis, and treatment can all benefit from the information that a medical image can offer. Due to the increased usage of digital medical imaging, numerous researchers are actively creating medical image processing algorithms and systems to provide the clinical community with improved results, such as accurate clinical parameters or processed images from the original images. We describe a Web-based DICOM reader in this work that was created solely using web technology, specifically Python, Streamlit, and Docker. When it comes to programming languages, Python has established itself as a competitor to MATLAB and Julia, two prominent scientific programming languages. In this work, we examine the potential of Python and Docker as co-implementers of the DICOM viewer. A demonstration field is available to new users, for learning purposes. Three different types of image planes namely axial, coronal, and sagittal are also available, which helps doctors readily detect cancers. Additionally, we can adjust the image plane's threshold levels to suit our needs. We also provide a JSON version of the website for use in forthcoming research initiatives.**

Keywords – DICOM viewer, medical imaging, python, planes

I. INTRODUCTION

New opportunities for the application of health information systems have been created by recent advancements in Web technology. Remote access to medical pictures for visualization and clinical revision is a radiology deployment of such methods. Assertive healthcare solutions can now be developed in a pure Web context because of new application programming interfaces (APIs). Software programmers must overcome certain difficult issues when creating these kinds of solutions, despite the HTML5 APIs' tremendous characteristics. If we intend to implement the system in a professional setting, access latency is a crucial criterion that cannot be overlooked. Several visualization techniques have been developed to aid in the interpretation of images. However, because these programs are designed for certain operating systems, their widespread adoption in expanding web-based working environments is hampered. Web applications that operate on browsers are widely used in many commercial sectors, including healthcare information systems. Because they are cross-platform, the user experience is the same regardless of platform. The Digital Imaging and Communications in Medicine (DICOM) standard governs how medical imaging data is handled, saved, printed, and transmitted. These DICOM pictures must be viewed and examined using a DICOM viewer tool. However, there are a number of challenges with medical imaging viewing. A common challenge is creating and administering DICOM-based applications with the unique transmission, decoding, and visualization needs. The business logic of the application must be kept apart from the graphical user interface. Additionally, the platform supports several mobile devices running IOS and Android operating systems. This choice entails numerous significant failure risks, which would squander several months of effort.

II. LITERATURE REVIEW

It is easy to see the value of DICOM viewers for doctors in the field of radiology. Numerous electronic platforms have been created since the dawn of the digital medical imaging era to offer users capabilities for processing and visualization. Over the past ten years, a number of web-based medical applications have been released. The value of DICOM viewers in supporting quick and effective processes has been amply shown in several papers.

Electronic tools that let radiologists search, retrieve, and view medical image studies from a DICOM archive enhance their daily activities. This is certainly not a reality. Arenson et al. [2] showed how personal computers might become highly helpful in teleradiology circumstances. They installed affordable medical image viewers on the PC and used a repository to view Imaging tests.

Eversman et al. [5] prepared a clinical viewing system was included in an electronic medical record (EMR) enabling the merging of DICOM images with textual reports. Additionally, they looked into the needs of the users and selected a few crucial characteristics for the viewer. They evaluated the system's performance and dependability last.

In order to be platform neutral, the first DICOM Web viewers were built using Java applets. However, because they need a special version of the Java Runtime Environment installed beforehand, those viewers are not regarded as pure web applications. There have been many more web technologies used.

In order to develop 3D volume rendering in the DICOM Web viewer, Mahmoudi et al. [15] looked into this. They provided 3D features over the Web by utilizing VRML (Virtual Reality Modeling Language). Meanwhile, WebGL API, which was included in HTML5 specs, has rendered VRML obsolete and made 3D representation possible.

Fern'andez-Bay'o et al. [6] proposed a technique utilizing a web server that permitted the inquiry and retrieval of images stored in a DICOM format. A little Java (Java) program called the DICOM Java viewer was executed in the browser and can be used to see the images using a web browser. The system's relatively basic graphical user interface supported primarily computed tomographic images and magnetic resonance imaging. Independent of its hardware and operating system, access was made possible through the use of a web browser user interface. The program was built using multi-platform intranet technologies including Java and a client-server intranet architecture. The systems slow reaction time when requesting photos from the web browser was by far its worst drawback.

Mata et al. [16] presented a similar web-based medical application to streamline clinical work by enabling the exchange of patient records across specialists. A countrywide exchange of patient records for a specific condition was the goal of this effort. Using an automated solution has several benefits, including standardized records and information communication. The web application was created using Java technology, and it used the SQL Language to connect to a web server and a patient database (SQL). Scalable designs, fit for the requirements of health care units with different sizes, are made possible by the usage of Java. Although the application runs on standalone machines, this means that the database server must also be installed on the same workstation. Because more potent database servers and workstations weren't available, these issues occurred.

Ortega et al. [17] suggested a web-based system for retinal image analysis, presenting a framework for ophthalmologists and other specialists in the area to collaborate utilising retinal image-based applications. The Arteriolar-to-Venular Ratio, a semi-automatic approach for measuring retinal microcirculation, is computed (AVR). Additionally, Remeseiro et al. [18] created another helpful web application for ophthalmologists in 2016 to detect dry eye illness by analyzing tear film photos. Tools for image processing and visualization are present in the majority of the currently available applications. Others, as the created by Kaspar et al. [9], however, put a greater emphasis on enhancing accessibility in real-time and enabling a remote and interactive stereoscopic visualization among many viewing locations. Lajara et al. [13] conducted research on a related topic in 2019 and came up with a method to deal with the absence of a standardized whole slide image (WIS) format, since it can communicate and operate with any WSI that adheres to the DICOM standard.

A web-tool was created to facilitate diagnostic clinical trials involving several experts, hospitals, or research facilities. This project's picture analysis is based on bone X-ray imaging, which also enables the database to store data and images. A similar project, MyCases, a portable database server for quick and simple storage of such cases comprising complete image series, patient data, and annotations, was introduced by Belgacem et al.[8].

A web-based, interactive, expandable, 2D and 3D medical image processing and visualization tool with no client installation was created by Mahmoudi et al [15]. Additionally, a variety of open source libraries-implemented medical picture preprocessing, registration, and segmentation techniques are included. Tiff, gif, jpg, and DICOM are among the image formats that the software supports. Young et al. [11] suggested a different strategy that included segmentation tools. This strategy used a web-based automatic spine segmentation method that used deep learning and might be highly useful and accurate for spine segmentation as a diagnostic procedure. Additionally, Yang et al. [20] investigated a web-based automated brain metastases segmentation and labelling platform to support the clinical workflow of stereotactic radiosurgery. Along with the segmentation, this also provided facilities for the doctor to manually annotate the data before exporting it locally or to a database.

Kim et al. [10] advocated the use of deep learning in yet another web application for diagnosing orthodontic issues. A fully automated cephalometric analysis method was to be created. Through the client webpage, the user can change the expected landmarks, and the database server saves the updated data.

By offering a series of image processing techniques that improve picture viewing and a set of drawing tools, used to annotate the problematic regions, Mata et al. [16] developed a web-app in 2012 to help diagnose new mammographic cases. Similar to this, Chen et al. [3] proposed a web-based system for CT image annotation in 2019 to support the development of a more reliable and accurate nodule detection system.

A user-friendly software for medical students that is compatible with the majority of Internet browsers and tablets was created by Colucci et al. in 2016 [3]. The software enables simultaneous display of photos in three planes and labelling of the images. Yuan et al. [21] evaluated RayPlus, a web-based platform for displaying and processing medical images that supports DICOM files, another helpful tool for radiologists. Processing medical images from any device is possible, and collaborative analysis is made simpler.

As can be seen from the work discussed above, medical imaging centers' software is attempting to advance and make use of these services as healthcare systems continue to leverage the internet and cloud technology more and more. Over the past ten years, there has been a considerable increase in the size of this field of study. Experts may now access and process medical photos from any device with the use of web applications, which also encourage expert collaboration. In our proposed methodology, we have used Docker to access the database and Python with a streamlit as front-end UI. Numerous sophisticated features were additionally included because we must adopt new technologies. A separate JSON format file for each distinct medical picture processing is displayed at the end and can be used in the future research process.

Table 1 shows the summary of web-based applications over the last decade. It includes the languages and image format used by each author.

Table 1. Summary of web-based applications over the last decade

Author	Year	Title	Database	Image format	Language	Web-based	Annotation system	Management
Mahmoudi	2010	Web-based interactive 2D/3D medical image processing and visualization software	n.d	Tiff, gif, jpeg, dicom	JS	✓	X	Data is stored in the user computer
Ortega	2010	Sirius: A web-based system for retinal image analysis	Retinal	n.d	JS, C++	✓	X	Data is stored in the user computer
Baltasar	2011	Design of a web-tool for Diagnostic Clinical Trials Handling Medical Imaging Research	Skeleton bones	Dicom	C++	✓	X	Data is stored
Mata	2012	MammoApplet: an interactive Java applet tool for manual annotation in medical imaging	Mammography	Dicom	JS	✓	✓	Data stored in the database
Gerstmair	2012	Intelligent image retrieval based on radiology reports	n.d	Dicom	JS	✓	X	Data can be downloaded
Kaspar	2013	An optimized web-based approach for collaborative stereoscopic medical visualization	n.d	Jpeg	C++	✓	X	App only allows visualization
Kammerer	2014	MyCaes- A portable application for radiologic case collections	n.d	Dicom	PHP	✓	X	Data is stored, but cannot be downloaded
Remeseiro	2016	iDEAS: a web-based platform for medical image processing	Eye	n.d	C++, JAVA	✓	✓	Data stored in the database
Colucci	2016	Development and utilization of a web-based application as a robust radiology teaching tool(RadStx) for medical student anatomy testing	n.d	Jpeg	JS	✓	✓	Data is stored
Yuan	2017	RayPlus: A web-based platform for medical image processing	n.d	Dicom	C++	✓	X	Uploading and downloading are disabled
Dumast	2018	A web-based system for neutral network based classification in temporomandibular joint osteoarthritis	Mandibular condyles	n.d	Python, C++	✓	X	Data can be downloaded
Lajara	2019	Optimum web viewer application for DICOM whole slide image visualization in anatomical pathology	n.d	Dicom	n.d	✓	X	Data is stored
Chen	2019	DeepLNAnno: a web based lung nodules annotating system for CT images	Lung	Dicom	n.d	✓	✓	Data export
Young	2020	Web-based spine segmentation using deep learning in computed tomography images	n.d	Jpeg	Python	✓	X	Data is stored in the server
Yang	2020	A web-based brain metastases segmentation and labeling platform for stereotactic radiosurgery	Brain	Dicom	JS	✓	✓	Data stored in the database
Kim	2020	Web-based fully automated cephalometric analysis by deep learning	Cephalograms	jpg	Python, Java	✓	✓	Data is stored in the server

III. Tools and Technique

A. Python

Image processing enables us to change and edit thousands of photos at once and gain insightful information from them. In practically every field, it has a wide range of uses. For this, Python is one of the most popular programming languages. Its incredible libraries and tools make it possible to complete the task of image processing extremely effectively.

B. Streamlit

The Streamlit framework is used by various machine learning engineers and data scientists to create user interfaces and potent machine learning applications from a trained model. By giving users interactive user interfaces, these applications can be used for visualization. To satisfy the requirements of your application, they offer an easier way to create charts, tables, and other figures. They also produce a prediction using the models that have been saved or selected in the app.

C. Docker

In general, Docker is a toolbox that enables developers to create, deploy, run, update, and stop containers using intuitive commands and labor-saving automation through a single API. Containers give developers the ability to bundle an application with all of its specifications and customizations, including libraries and other dependencies, and deliver it as a single package. Developers may develop containers without Docker, but the platform makes it simpler, safer, and easier to design, deploy, and manage containers.

a. Working of Docker

Client-server architecture is utilized by Docker. The Docker client communicates with the Docker daemon, which handles the labor-intensive tasks of creating, executing, and distributing your Docker containers. Both the Docker client and daemon can run on the same machine, or a Docker client can be connected to a remote Docker daemon. UNIX sockets, a network interface, or a REST API are all used by the Docker client and daemon for communication. Docker Compose is an additional Docker client that enables you to deal with applications made up of a collection of containers.

b. Advantages of Docker in Image Processing

- Rapid and simply configured
- Application segregation
- Increase in performance.
- Swarm
- Scheduling Mesh
- Security administration
- Quick machine proliferation

D. Streamlit cloud

A public cloud framework called Streamlit cloud has recently gained in popularity. Due to its ease of use, it is one of the most preferred options for many development projects. It facilitates the easy building and deployment of applications with a focus on enabling customer-focused apps. Because the Streamlit cloud platform manages hardware and servers rather than the infrastructure that supports them, businesses using it can concentrate on perfecting their apps.

IV. Implementation

There are two basic ways to launch this process and application:

- Either the user installs all the requirements, clones the GitHub repository on his PC, and executes the program in a local environment.
- Or the user can access the program directly through a dedicated URL without having to always install it.

We have made the decision to deploy the application on Streamlit cloud (A Cloud Platform) and make it accessible via a single URL in a secure environment in order to make the application available to the largest number of people and consequently people who are not necessarily used to dealing with GitHub.

V. Setup of the Environment

We have setup the environment, which can be accessed from both Docker as well as anaconda command prompt.

A. Docker

- Clone the repository and set the current directory:

```
git clone
cd dicom-plane-viewer/
```

- For running a local docker container, change the line in the Docker file from:

```
CMD streamlit run DICOM.py --
server.port $PORT
                    to
CMD streamlit run DICOM.py --
server.port 8501
```

- Then, you can build and run successfully the container with:

```
docker build ./ --tag webapp:v1
docker container run -p 8501:8501
webapp:v1
```

- To finish, open the browser at http://localhost:8501/

B. Conda Environment

- Make sure you have Anaconda installed since it is the easiest way to setup GDCM on Python3 which is a requirement for the pydicom library.
- (Optional) Create a conda environment for installing and running the app:

```
conda create --name DICOM_env
python=3.6.10 -y
conda activate DICOM_env
```

- Clone the repository and set the current directory:

```
git clone
cd dicom- plane-viewer/webapp/
conda install -c conda-forge -y gdcm
pip install -r requirements.txt
streamlit run DICOM.py
```

- To finish, open the browser at http://localhost:8501/

VI. ARCHITECHTURE

The main objective of the project is to create an application that can be used in a web-based environment. As a result, we decided against creating a standalone version because doing so necessitates a time-consuming installation process and additional programming.

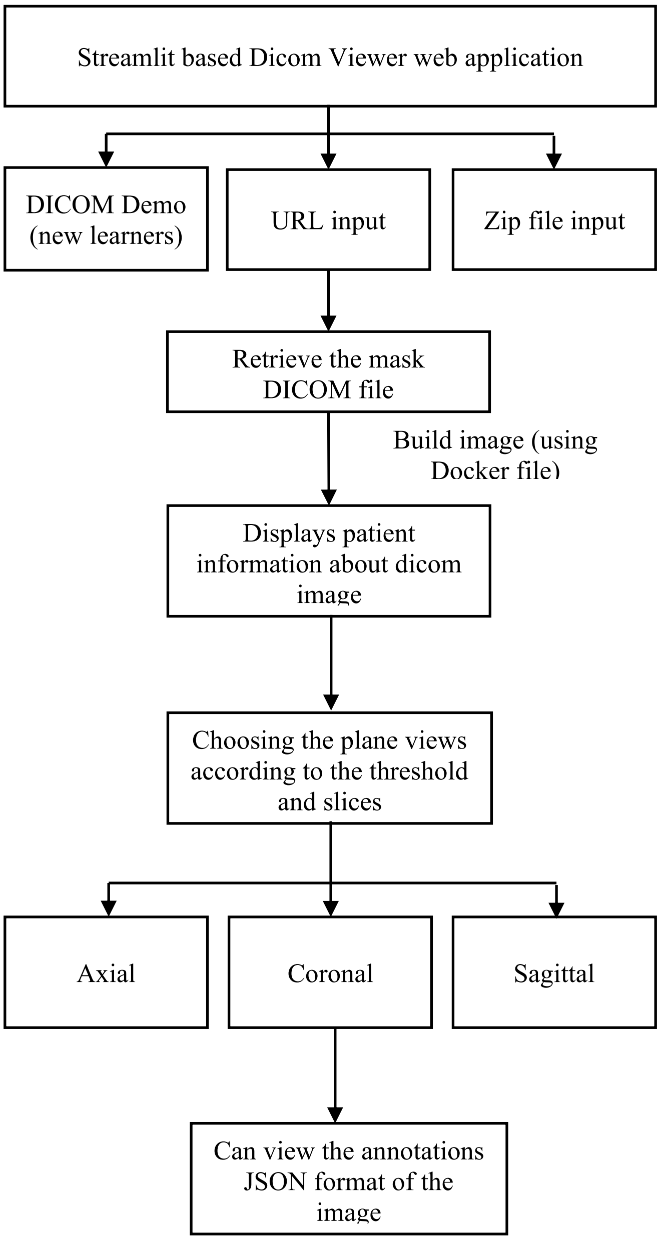

Figure 1. Workflow of Proposed methodology

For the above said reason, a Python Streamlit interface that would have made it simpler to create these kinds of apps was abandoned. We sought develop a more flexible tool, where people could access the web and run applications without having to download cumbersome software. The main cause of this is that many hospitals and medical facilities lack the necessary infrastructure to handle these intricate installations. In order to avoid the necessity for installations, we decided to adopt an alternative approach. We used Streamlit to create the user interface and Docker as a HUB repository to deploy our web-based application.

VII. WORKING

A platform for collaboration and version control is called GitHub. We set up a repository where all of the project's files can be found.

The DICOM.py file contains the primary code. In order to execute the project, you must first install all of its dependencies by using the command pip install -r requirements.txt. The app will then start running once you issue the command python DICOM.py (or python3 app.py, depending on the version of Python you have installed on your system).

The viewer has a feature that allows to mark the slices of a certain set of Dicom files that contain errors. Each series' annotations have the labels "Anomaly" and "Slices" and are in the JSON format. The program only works with zip files that have one or more folders, which are typically shown as series and contain Dicom files. The example folder contains a sample of a valid zip file that may be examined. Zip files can be submitted using the file upload widget or publicly shared URLs from Google Drive. Uploads of zip files are restricted to 100MB because the demo is hosted on Streamlit cloud, whose free tier dyno has a meagre memory allocation. Table 2 shows the summary of the key points of our application. Figure 2, 3, 4 shows the Axial, Coronal, Sagittal views of the image.

Table 2. Summary of the key points of our application

Programming language	Python
Architecture	Web-based (Streamlit)
Annotations system	Export and import medical annotations
Repositories and deployment	GitHub and Docker server
Materials	DICOM database
Planes	Axial, Sagittal, Coronal
Input type	URL, Zip files

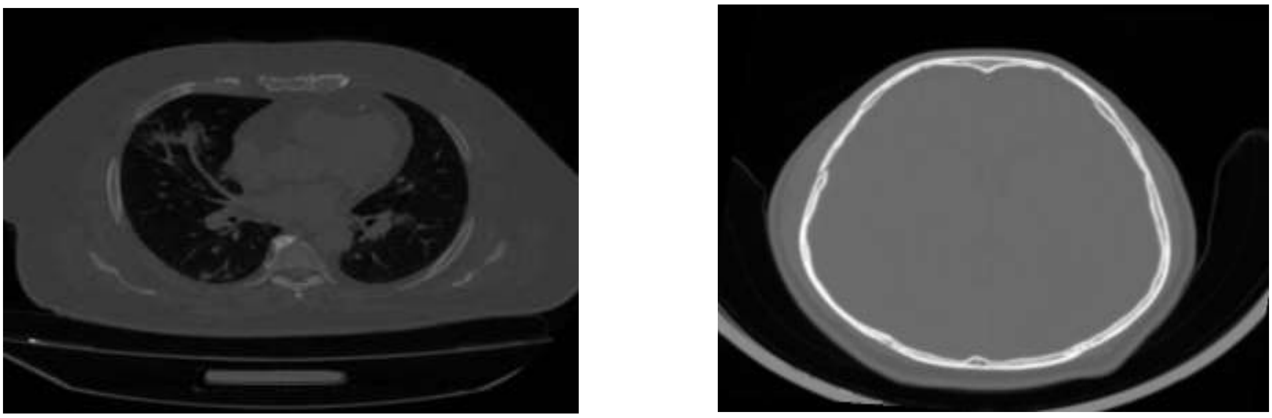

Figure 2. Axial View

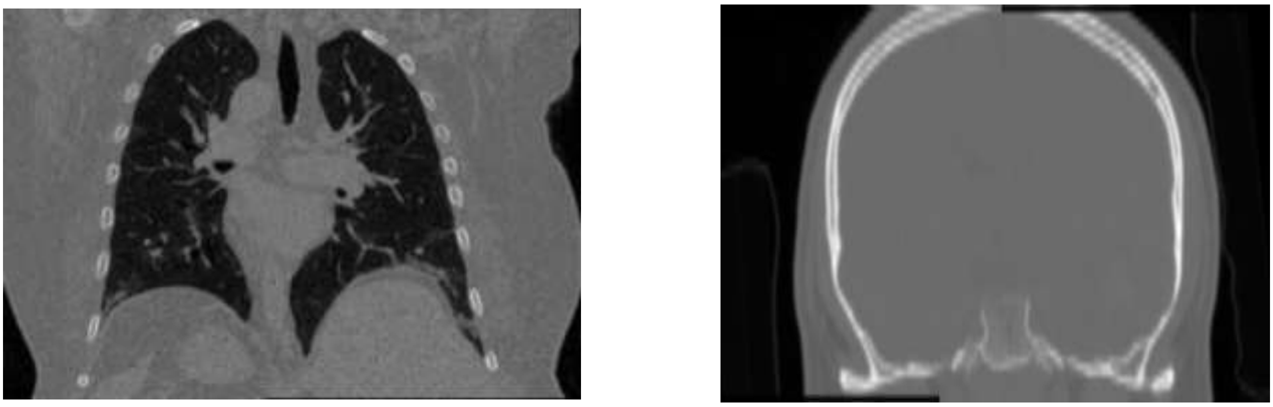

Figure 3. Coronal View

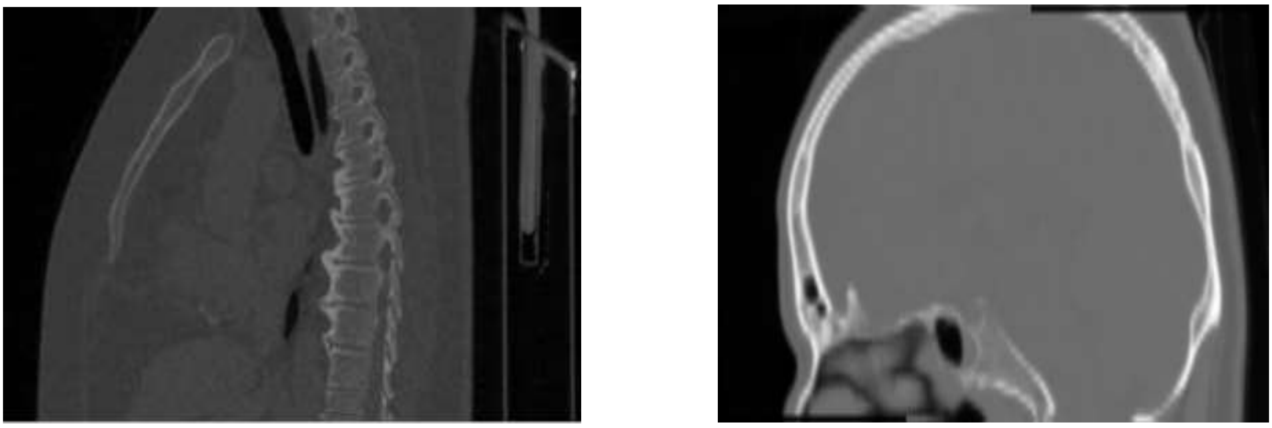

Figure 4. Sagittal View

VIII. CONCLUSION

Our main objective was to create a medical tool that would help the medical community manage cancer while showcasing the benefits and potential of using a web-based application. Thus, the creation of this project has been influenced by the significant trend of incorporating new technologies into the software of medical imaging centers. It comprises the DICOM data table, the image viewer that displays three separate plane views, the system for making annotations, and the export and import options for these discoveries. The program's source code is accessible on GitHub, where it can be downloaded and executed using a Python interpreter via Anaconda. Additionally, it can also be used with Docker. To implement, all that is required is to access the Docker Hub and open the repository for DICOM Viewer. The Docker software is simple and cost-free to download. Installing a complicated system structure is therefore not required. In conclusion, the first version of DICOM Viewer that was produced achieves all of the initial particular and general objectives. With the assistance of medical professionals and hospitals, we created a tool that would be utilized by the medical community to diagnose cancer. Although this first version is effective and ready for use, we have considered certain features that could enhance its use while it was being developed.

References

[1] A. Golubev, N. Iliuha and P. Bogatencov, "“DICOM Network” services DICOM data exchange solution integrated in the regional VI-SEEM infrastructure," IEEE EUROCON 2017 -17th International Conference on Smart Technologies, 2017, pp. 558-563.

[2] Arenson RL, Andriole KP, Avrin DE, Gould RG. Computers in imaging and health care: now and in the future. J Digit Imaging. 2000 Nov;13(4):145-56.

[3] Chen, S., Guo, J., Wang, C., Xu, X., Yi, Z., & Li, W, (2019), DeepLNAnno: a Web-Based Lung Nodules Annotating System for CT Images. Journal of Medical Systems, 43(7)

[4] Colucci, P. G., Kostandy, P., Shrauner, W. R., Arleo, E., Fuortes, M., Griffin, A. S., Huang, Y. H., Juluru, K., & Tsiouris, A. J, (2015), Development and Utilization of a Web-Based Application as a Robust Radiology Teaching Tool (RadStax) for Medical Student Anatomy Teaching. Academic Radiology, 22(2), 247–255

[5] Eversman, William G. et al. “Performance and function of a desktop viewer at mayo clinic scottsdale.” Journal of Digital Imaging 13 (2009): 147-152.

[6] Fernandez-Bayo, J. et al. “Distributing medical images with internet technologies: a DICOM web server and a DICOM java viewer.” Radiographics : a review publication of the Radiological Society of North America, Inc 20 2 (2000): 581-90 .

[7] Guld, Mark & Kohnen, Michael & Keysers, Daniel & Schubert, Henning & Wein, Berthold & Lehmann, Thomas. (2002). Quality of DICOM header information for image categorization. Proceedings of SPIE - The International Society for Optical Engineering. 4685. 10.1117/12.467017.

[8] K. Belgacem, M. Kenoui, F. Bouguerra, M. Laidi, A. Semrani and C. Sellah, "Collaborative Visualization and Annotations of DICOM Images for Real-Time Web-based Telemedicine System," 2021 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI), 2021, pp. 1-6

[9] Kaspar, M., Parsad, N. M., & Silverstein, J. C. (2013). An optimized web-based approach for collaborative stereoscopic medical visualization. Journal of the American Medical Informatics Association, 20(3), 535–543.

[10] Kim, H., Shim, E., Park, J., Kim, Y. J., Lee, U., & Kim, Y, (2020), Web-based fully automated cephalometric analysis by deep learning. Computer Methods and Programs in Biomedicine, 194, 105513

[11] Kim, Y. J., Ganbold, B., & Kim, K. G, (2020), Web-Based Spine Segmentation Using Deep Learning in Computed Tomography Images. Healthcare Informatics Research, 26(1), 61

[12] L. Rebelo, F. Fidalgo and Â. Oliveira, "Radiology Information System (RIS)," 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), 2022, pp. 1-6.

[13] Lajara, N., Espinosa-Aranda, J. L., Deniz, O., & Bueno, G. (2019). Optimum web viewer application for DICOM whole slide image visualization in anatomical pathology. Computer Methods and Programs in Biomedicine, 179, 104983.

[14] M. Mustra, K. Delac and M. Grgic, "Overview of the DICOM standard," 2008 50th International Symposium ELMAR, 2008, pp. 39-44.

[15] Mahmoudi, S. E., Akhondi-Asl, A., Rahmani, R., Faghih-Roohi, S., Taimouri, V., Sabouri, A., & Soltanian-Zadeh, H, (2010), Web-based interactive 2D/3D medical image processing and visualization software. Computer Methods and Programs in Biomedicine, 98(2), 172–182

[16] Mata, C., Oliver, A., Torrent, A., & Marti, J, (2012), MammoApplet: An interactive Java applet tool for manual annotation in medical imaging. 2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE)

[17] Ortega, M., Barreira, N., Novo, J., Penedo, M., Pose-Reino, A., & Gómez-Ulla, F. (2010). Sirius: A web-based system for retinal image analysis. International Journal of Medical Informatics, 79(10), 722–732

[18] Remeseiro, B., Barreira, N., García-Resúa, C., Lira, M., Giráldez, M. J., Yebra-Pimentel, E., & Penedo, M. G. (2016). iDEAS: A web-based system for dry eye assessment. Computer Methods and Programs in Biomedicine, 130, 186–197.

[19] Y. Gupta, C. Costa, E. Pinho and L. A. Bastião Silva, "A DICOM Standard Pipeline for Microscope Imaging Modalities," 2021 IEEE Symposium on Computers and Communications (ISCC), 2021, pp. 1-6

[20] Yang, Z., Liu, H., Liu, Y., Stojadinovic, S., Timmerman, R., Nedzi, L., Dan, T., Wardak, Z., Lu, W., & Gu, X. (2020). A web-based brain metastases segmentation and labeling platform for stereotactic radiosurgery. Medical Physics, 47(8), 3263–3276.

[21] Yuan, R., Luo, M., Sun, Z., Shi, S., Xiao, P., & Xie, Q, (2016), RayPlus: a Web-Based Platform for Medical Image Processing. Journal of Digital Imaging, 30(2), 197–203