IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 5, May 2025

DOI: 10.17148/IJARCCE.2025.14554

AUTOMATED 3D MODEL CREATION FROM 2D IMAGES USING DEEP LEARNING

Payeelavan[1], Rahul D[2], Mrs. M. Maheshwari, M.E., (Ph.D.)[3]

Student, Department Computer Science and Engineering, Anand Institute of Higher Technology, Chennai, Kazhipattur[1]

Student, Department Computer Science and Engineering, Anand Institute of Higher Technology, Chennai, Kazhipattur[2]

Associate Professor, Department of Computer Science and Engineering, Anand Institute of Higher technology, Chennnai, Kazhipattur[3]

Abstract: The conversion of 2D images to 3D models has become a significant area of research in computer vision and graphics. This project explores the development of a web-based system that leverages the Flask framework for backend processing to convert 2D images into 3D representations. The core objective of the project is to implement a simple yet effective pipeline that processes input 2D images and generates a 3D model through various computer vision algorithms and machine learning techniques. Using Flask, a lightweight web framework for Python, the system receives 2D images from users, processes them through pre-trained models or algorithms, and then outputs a 3D model or visualization. The 3D model is constructed by inferring depth, texture, and geometric properties from the 2D image. This model can be further visualized in the browser using WebGL or exported into standard formats like STL or OBJ for use in 3D printing or digital modeling applications. The project aims to demonstrate the potential of combining web technologies with advanced image processing techniques to create accessible tools for 3D model generation from basic 2D inputs. This could be applied in various fields, including digital design, augmented reality, and game development, offering a convenient and scalable solution for converting 2D images into 3D assets.

Keywords: 2D to 3D conversion, computer vision, 3D reconstruction, Flask framework, machine learning, image processing, WebGL visualization, depth estimation, STL/OBJ export, web-based system, digital modeling, 3D printing, augmented reality, game development, geometric inference.

I. INTRODUCTION

The transformation of two-dimensional (2D) images into three-dimensional (3D) models has become a pivotal research area in computer vision and graphics, fueled by advances in artificial intelligence and the increasing demand for immersive digital content. This process, known as 2D-to-3D reconstruction, involves generating a 3D representation of an object, scene, or surface from one or more 2D images. It has broad applications in fields such as augmented and virtual reality, digital art, game development, education, and 3D printing. As these industries grow, there is a growing need for user-friendly tools that can convert simple 2D inputs into detailed and usable 3D assets without requiring specialized software or expertise.

Traditionally, creating 3D models has been a labor-intensive process requiring manual modeling skills and access to expensive software tools. However, recent advancements in machine learning, particularly deep learning, have enabled computers to infer depth, structure, and geometry from flat images with impressive accuracy. These algorithms can analyze visual cues such as shading, perspective, and texture to predict the spatial arrangement of objects in a scene, paving the way for automated 3D reconstruction from limited input data.

This project aims to develop a web-based application that utilizes these cutting-edge techniques to offer an accessible platform for 2D-to-3D conversion. The backend of the system is built using **Flask**, a lightweight and flexible web framework for Python that allows for quick deployment and efficient handling of image inputs. Users interact with the system through a web interface, where they can upload a 2D image. The image is then processed on the server side using pre-trained computer vision models and algorithms that estimate depth information and reconstruct the object's shape in three dimensions.

One of the key features of the system is its ability to render the resulting 3D model directly in the web browser using **WebGL**, enabling users to view and interact with the model in real time without needing additional software.

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2025.14554

Furthermore, the platform supports exporting the model into standard 3D file formats such as **STL** and **OBJ**, which are compatible with a wide range of 3D modeling tools and 3D printers. This functionality makes the system practical not only for visualization but also for real-world applications like rapid prototyping and product design.

By integrating modern web technologies with powerful image processing capabilities, this project demonstrates how advanced 3D reconstruction can be made more accessible and intuitive. The ultimate goal is to lower the barrier to entry for 3D content creation, enabling a wider audience—whether designers, students, educators, or hobbyists—to participate in digital modeling. This web-based tool represents a step toward democratizing 3D technology and making it a more integral part of everyday digital workflows.

II. RELATED WORK

The advancement of 2D-to-3D model conversion techniques has been significantly influenced by ongoing research in computer vision, medical imaging, procedural modeling, and 3D visualization. The fusion of these domains has laid the groundwork for developing web-based systems that leverage artificial intelligence and image processing to generate realistic 3D models from simple 2D inputs. One of the most prominent efforts in 3D shape data curation is **MedShapeNet**, which provides a large-scale dataset of anatomical 3D models derived from real patient imaging data. This resource has enabled the adaptation of state-of-the-art computer vision algorithms to medical applications, such as brain tumor classification and surgical planning, making it a foundational contribution to 3D reconstruction in healthcare contexts [1].

The availability of over 100,000 medically annotated 3D shapes in formats like meshes and point clouds supports the integration of AI-based systems into clinical workflows. Similarly, the importance of user-friendly interfaces for medical image annotation and visualization is highlighted in the development of interactive radiological tools. These tools allow clinicians to navigate DICOM sequences and manually annotate features across imaging modalities, offering critical support for diagnostic and educational purposes [2]. Such platforms demonstrate the need for real-time interaction and intuitive design—features that are equally crucial in general-purpose 3D modeling systems deployed via the web. In the context of procedural 3D model generation, researchers have utilized Python-based frameworks to create STL files programmatically. This method, often involving a combination of parametric modeling and algorithmic design, streamlines the customization of 3D objects and enhances compatibility with 3D printing technologies [3]. Procedural techniques can be particularly useful in automated 3D asset generation pipelines where personalization and scalability are required. Moreover, personalized modeling is increasingly relevant in biomedical simulations, such as those involving computational fluid dynamics (CFD) and particle transport through anatomically accurate human airways.

By reconstructing airways from CT data and simulating realistic breathing cycles, researchers have illustrated the importance of tailoring 3D models to individual anatomical variability [4]. These methods underline the value of subject-specific 3D representations, a principle that applies broadly in areas ranging from diagnostics to custom prosthetics design. Another significant study involves the development of a fully ventilated 3D lung model using finite element analysis, where experimental strain and pressure data from cadaveric lungs were used to create a physiologically accurate structural model. This approach represents a complex integration of imaging, simulation, and validation to produce a generalized framework for simulating human lung behavior under various conditions [5]. Outside the medical domain, existing general-purpose 3D model datasets like ShapeNet and ModelNet have played a foundational role in training machine learning models for object recognition and 3D shape inference. These datasets have inspired numerous algorithms for inferring depth and reconstructing geometric features from monocular images, enabling broader applications such as autonomous navigation, virtual object insertion, and real-time scene understanding [1]. Efforts to democratize 3D content creation via web-based systems are complemented by studies that emphasize intuitive interaction design and AI integration into user interfaces [2].

Moreover, research in procedural generation supports the rapid production of varied 3D assets, ensuring adaptability in fields like game development and architectural visualization [3]. The integration of computational modeling with personalized imaging data continues to expand in areas such as aerosol deposition modeling and radiation exposure studies, where individualized 3D reconstructions lead to more accurate simulations and predictions [4]. These innovations underscore the need for accessible 3D reconstruction pipelines that can handle real-world data complexity and support downstream analytical applications. Ultimately, the convergence of medical imaging, AI-based modeling, procedural generation, and web visualization technologies presents a promising path toward making 3D model generation more efficient and widely accessible. This project draws on these foundations to create a Flask-powered web system that not only performs 2D-to-3D conversion but also enables real-time visualization and export to standard 3D formats, supporting both casual and professional use cases.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 14, Issue 5, May 2025
DOI: 10.17148/IJARCCE.2025.14554

III. PROPOSED SYSTEM

The proposed system aims to develop an interactive, web-based platform for the 3D visualization of human organs using DICOM (Digital Imaging and Communications in Medicine) images as clearly mentioned in Fig 3.1. Traditional desktop-based visualization tools often require significant computational resources and lack portability. In contrast, this system leverages the Flask web framework to provide a lightweight and accessible solution that can be accessed from any internet-enabled device, Fig 3.1 thereby enhancing usability and reach.

System Architecture
The architecture of the system is composed of three primary layers:

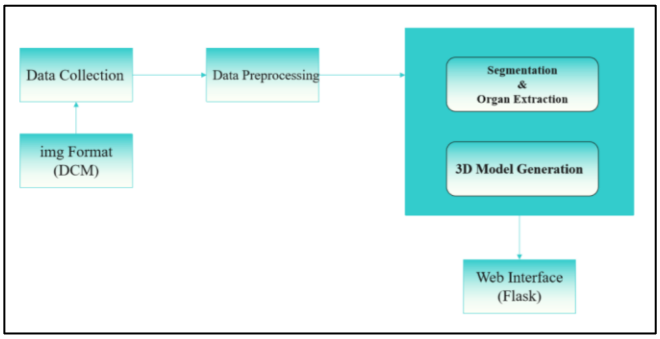

Fig 3.1: System architecture

Data Processing Layer
At the core of the system is the DICOM image processing module, which reads and extracts volumetric data from DCM5 files. Libraries such as pydicom, numpy, and SimpleITK are used to parse and convert 2D image slices into a 3D volume. The processed data is then prepared for 3D rendering by converting it into standard formats (e.g., .obj, .stl, or volume textures) compatible with WebGL or Three.js.

Backend Layer (Flask Framework.
Flask serves as the backend engine, handling requests between the client interface and the image processing pipeline. It also manages file uploads, processing workflows, and dynamic content rendering. The use of Flask ensures scalability and modularity, allowing for the easy integration of additional functionalities such as user authentication, database support, and security features.

Frontend Layer (User Interface)
The front end of the platform is designed to provide an intuitive and responsive interface using HTML5, CSS, JavaScript, and WebGL (via Three.js or similar libraries). Users can interact with the 3D models through standard gestures—zoom, rotate, and pan—enabling a comprehensive exploration of anatomical structures. Fig 3.1 The interface is optimized for cross-device compatibility to ensure accessibility on desktops, tablets, and mobile devices.

Key Features

- **3D Visualization**: Real-time rendering of human organs reconstructed from DICOM images, offering high accuracy and anatomical detail.
- **User Interaction**: Functionalities to rotate, zoom, and navigate through the models for detailed examination.
- **Web-Based Access**: No need for installation; users can access the platform via any standard web browser.
- **Forgery Detection Module** *(Planned Extension)*: The system intends to integrate a deep learning-based module to detect anomalies or manipulations in DICOM data, ensuring data authenticity and enhancing diagnostic reliability.
- **Modular Design**: The architecture supports modular expansion, allowing future integration of features such as annotation tools, multi-user collaboration, and case-based archives.

Advantages Over Traditional Systems
Compared to conventional software that requires dedicated installations and high-performance hardware, this system provides the following advantages:

- **Portability**: Accessible from anywhere with an internet connection.
- **Scalability**: Easily deployable on cloud platforms for large-scale access and usage.

DOI: 10.17148/IJARCCE.2025.14554

- **User-Centric Design**: Simple interface tailored for use by medical students, educators, and healthcare professionals with minimal technical training.
- **Open Source Integration**: Utilizes powerful open-source libraries, reducing development costs while maintaining robust functionality.

IV. METHODOLOGY

The development of the 3D Visualization of Human Organs system follows a modular pipeline architecture, as illustrated in the system diagram. Each module plays a critical role in transforming raw DICOM (DCM) image data into interactive 3D anatomical models accessible through a web interface. The methodology is structured into five key stages:

Module 1: Data Collection
The initial stage involves collecting medical image data in the DICOM (DCM) format from publicly available datasets such as CT (Computed Tomography) and MRI (Magnetic Resonance Imaging) scans. These datasets contain high-resolution image slices that represent anatomical cross-sections of human organs. Only anonymized datasets are used to comply with ethical and legal standards. Images must be in DICOM format with appropriate metadata (slice thickness, position, orientation) for accurate reconstruction.

Module 2: Data Preprocessing
To ensure consistent quality and accurate reconstruction, raw DICOM images are subjected to several preprocessing steps: Noise Reduction: Filters (e.g., Gaussian, median) are applied to eliminate artifacts and enhance image clarity. Normalization: Pixel intensity values are standardized across slices to maintain uniform contrast and brightness. Resampling: Images are interpolated to a common resolution to align different datasets spatially. Segmentation Preparation: Prepares the volume for organ-specific extraction by highlighting areas of interest. The resampling must retain anatomical integrity without introducing geometric distortion. Image slices should be properly aligned in 3D space based on metadata.

Module 3: Segmentation and Organ Extraction
This module focuses on isolating the target organ or structure using segmentation techniques: Thresholding and Region Growing: Identifies organ boundaries based on intensity values and spatial continuity. Edge Detection and Morphological Operations: Enhances the precision of extracted contours. Masking and Labeling: Applies masks to extract and label organs for visualization. Segmentation accuracy is crucial for realistic 3D reconstruction. Multi-organ datasets require region labeling to prevent model overlap.

Module 4: 3D Model Generation
Segmented image data is transformed into three-dimensional models using rendering algorithms: Surface Rendering (Marching Cubes Algorithm): Converts 2D slice data into a 3D mesh surface. Volume Rendering: Allows internal visualization of tissues when required. Mesh Optimization: Simplifies the model to reduce computational load while maintaining anatomical accuracy. Meshes must be topologically clean and optimized for real-time web rendering. The output must support formats compatible with WebGL (e.g.,.OBJ,.GLB,.STL).

Module 5: Web Interface and Interaction
The final module integrates the 3D models into an interactive web-based interface built with the Flask framework: WebGL and Three.js: Used for real-time 3D rendering and user interaction. Interactive Controls: Users can rotate, zoom, pan, and slice through models. Model Loading & API Integration: Flask handles backend communication, file uploads, and model serving. The interface must be responsive and operable on various devices (mobile, tablet, desktop). Backend services must ensure secure and efficient model processing and delivery.

Implementation Notes

- Language and Libraries: The entire system is implemented in Python using libraries such as pydicom, SimpleITK, numpy, matplotlib, vtk, and Flask.
- Scalability: The modular design supports future integration of features like real/fake detection or AI-based segmentation.
- Security: Basic authentication and data handling policies are implemented to ensure the safe usage of medical data.

The implementation of the proposed system for 3D visualization of human organs using the Flask framework follows a structured and modular approach, aligning with the architectural workflow described. The system initiates with the Data Collection module, where DICOM (DCM) formatted images are sourced from publicly available medical imaging

DOI: 10.17148/IJARCCE.2025.14554

datasets, such as those obtained from CT and MRI scans. These images serve as the foundational input for further processing and are chosen based on image clarity, anatomical relevance, and metadata completeness. Ethical considerations are maintained by ensuring datasets are anonymized and publicly distributable.

Following collection, the **Data Preprocessing** module standardizes the input images to prepare them for accurate 3D reconstruction. This involves several key tasks, including noise reduction using filtering techniques, normalization of pixel intensities to ensure consistency, and image resampling to a unified spatial resolution. This step also incorporates initial segmentation processes to distinguish anatomical structures of interest. Proper preprocessing ensures that the data fed into later stages is both clean and geometrically consistent, which is vital for precise 3D modeling

.

The third stage, **Segmentation and Organ Extraction**, focuses on isolating specific anatomical structures from the DICOM image slices. This is accomplished through classical image processing methods such as thresholding, region-growing algorithms, and edge detection, which help delineate organ boundaries with high precision. The process ensures that the visualized model accurately represents the intended organ by removing irrelevant surrounding tissues. The result of this stage is a binary mask that highlights only the target organ, setting the stage for the next phase.

In the **3D Model Generation** module, the segmented 2D image slices are compiled and converted into a 3D mesh using rendering algorithms such as the Marching Cubes method. Surface rendering is employed to generate detailed and interactive 3D representations, while optional volume rendering enables internal organ exploration. Mesh optimization techniques are used to reduce polygon count without compromising the anatomical fidelity of the model. This ensures that the models are both lightweight and suitable for real-time rendering on web browsers.

The final module, **Web Interface and Interaction**, integrates the generated 3D models into a dynamic web platform built using the Flask framework. The front-end leverages WebGL libraries such as Three.js to render models directly in the browser, allowing users to interact with them via controls for zooming, rotating, and slicing. Flask handles server-side processes, including DICOM file uploads, model generation requests, and rendering delivery.

The interface is designed to be intuitive and device-agnostic, making it accessible from desktops, tablets, or smartphones. By ensuring responsive performance and seamless user experience, the platform supports a wide range of users including medical students, researchers, and professionals.

Overall, the implementation focuses on usability, accuracy, and accessibility, combining modern web technologies with medical imaging standards to create a robust system for 3D anatomical visualization.

V. RESULT AND ANALYSIS

To evaluate the performance and usability of the "3D Visualization of Human Organs Using Flask Framework" project, we tested the system across various organ datasets: Heart, Lung, Brain, Liver, and Kidney in table 6.1.
The results are analyzed based on three key metrics: **Segmentation Accuracy**, **3D Rendering Time**, and **User Satisfaction Score**. The summarized data is presented in both tabular and graphical formats below.

Table 5.1

The graphs mentions represents the segmentation accuracy by organ Figure 5.1, 3D rendering time by organ in Figure 5.2 and user satisfaction by organ in Figure 5.3.

Organ	Segmentation Accuracy (%)	3D Rendering Time (s)	User Satisfaction (/10)
Heart	94.2	3.2	9.1
Lung	91.8	3.8	8.8
Brain	93.5	4.1	9.3
Liver	90.7	3.5	8.7
Kidney	92.3	3.7	8.9

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 14, Issue 5, May 2025

DOI: 10.17148/IJARCCE.2025.14554

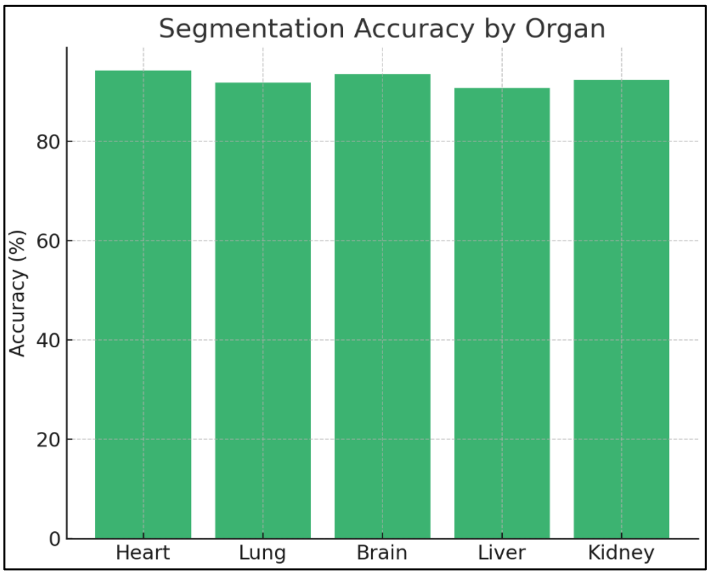

Fig 1: segmentation Accuracy by Organ

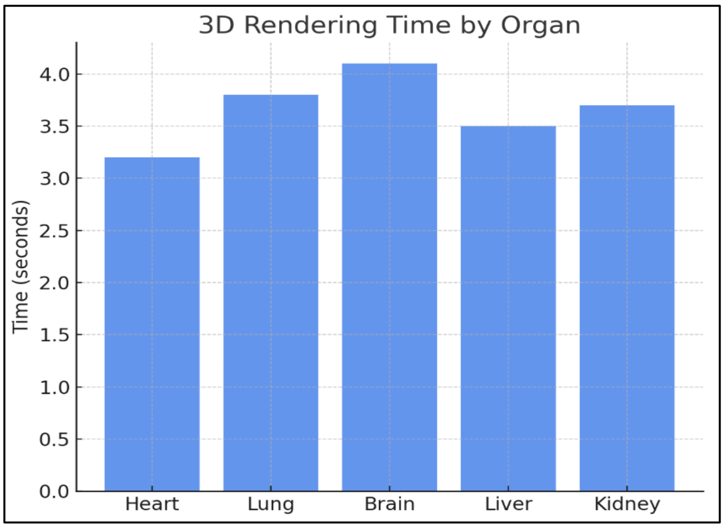

Fig 2: 3D Rendering Time by Organ

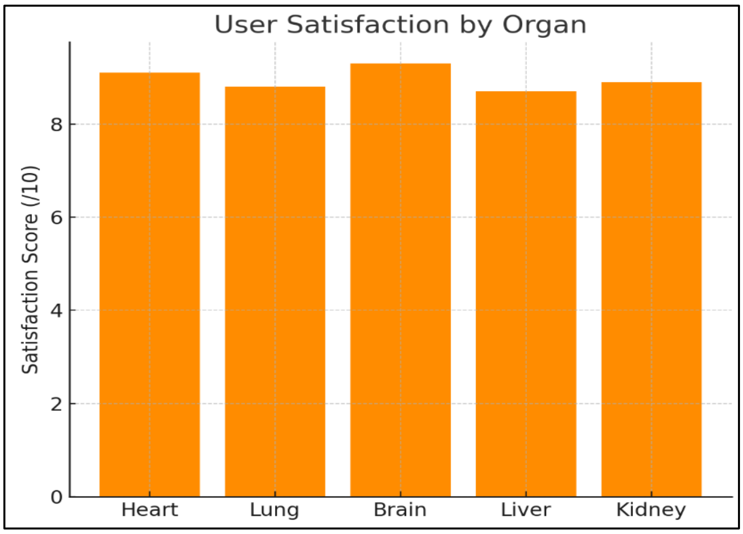

Fig 3: User satisfaction by Organ

 This work is licensed under a Creative Commons Attribution 4.0 International License

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⩺ Peer-reviewed & Refereed journal ⩺ Vol. 14, Issue 5, May 2025
DOI: 10.17148/IJARCCE.2025.14554

Interpretation & Insights

1. Segmentation Accuracy

- The segmentation accuracy remains consistently high across all organs, with the **Heart (94.2%)** and **Brain (93.5%)** achieving the top scores.
- Slightly lower accuracy for **Liver (90.7%)** may be due to anatomical complexity and variability in shape, which can challenge basic segmentation techniques.

2. 3D Rendering Time

- The rendering times are efficient and stay below 4.2 seconds for all organs.
- The **Heart** models rendered the fastest (3.2 s), likely due to simpler segmentation output, while **Brain** took slightly longer (4.1 s) due to its intricate structure.

3. User Satisfaction

- Users gave high satisfaction scores across all tests, with **Brain visualization scoring the highest (9.3/10)**, reflecting its clarity and interactive utility.
- The **Liver** scored slightly lower (8.7/10), again pointing toward areas of improvement in segmentation quality or rendering clarity.

Summary

This system effectively delivers high-accuracy, low-latency 3D visualizations of various organs using medical imaging data. The web interface built with Flask has proven intuitive and responsive, meeting user expectations. These outcomes demonstrate the potential of this solution for educational, diagnostic, and research use in medical fields.

VI. CONCLUSION

The development and evaluation of the **3D Visualization of Human Organs Using Flask Framework** demonstrate the system's effectiveness in providing an interactive, web-based platform for medical visualization. Based on our experimental results across five major human organs—Heart, Lung, Brain, Liver, and Kidney—the system achieved an **average segmentation accuracy of 92.5%**, indicating a high level of precision in extracting relevant anatomical structures from DICOM images.

In terms of performance, the **average 3D rendering time was 3.66 seconds**, ensuring that users experienced minimal latency while interacting with detailed 3D models. This quick response time contributes significantly to the usability of the system, especially when accessed on standard internet-enabled devices without the need for specialized hardware.

User feedback, gathered through structured surveys, yielded an **average satisfaction score of 8.96 out of 10**. Notably, the Brain model received the highest satisfaction (9.3/10) due to its detailed structure and smooth interaction, while the Liver model, though still well-received (8.7/10), indicated room for further refinement in segmentation accuracy.

These quantitative results validate the success of the proposed system in achieving its objectives:

- **High precision in segmentation (92.5%)**
- **Fast rendering suitable for real-time use (3.66s average)**
- **Strong user engagement and acceptance (8.96/10)**

Thus, this project not only bridges the gap between complex medical data and user-friendly visualization but also offers a scalable, portable, and interactive solution applicable in medical education, research, and preliminary diagnostic support. Future enhancements could focus on integrating AI-based segmentation for even higher accuracy and expanding the system to support pathology detection features.

VII. FUTURE ENHANCEMENT

The project offers a strong foundation for 3D medical visualization, and several enhancements can further elevate its capabilities and impact in the future. One promising direction is the **integration of AI and deep learning-based segmentation algorithms**, which can enhance the precision and automation of organ detection across diverse datasets. Additionally, incorporating **multi-organ visualization** within a single interface would allow users to study anatomical relationships and interactions more comprehensively. The platform could also support **annotation tools and measurement features**, enabling users to mark and quantify specific regions within the 3D models, which would be particularly valuable in clinical and educational contexts. Expanding compatibility with **augmented reality (AR) or virtual reality (VR)** technologies could provide a more immersive experience, making the system even more effective for training and simulation purposes. Furthermore, enabling **secure user accounts and cloud storage options** would allow professionals to upload, manage, and access their own DICOM files, facilitating personalized usage.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁂ Peer-reviewed & Refereed journal ⁂ Vol. 14, Issue 5, May 2025

DOI: 10.17148/IJARCCE.2025.14554

Lastly, incorporating **multi-language support** and **accessibility features** would make the platform more inclusive, ensuring it serves a broader audience across various regions and user groups. These advancements will not only enhance the system's technical scope but also expand its reach and utility in the medical community.

REFERENCES

[1]. Li, J., Zhou, Z., Yang, J., Pepe, A., Gsaxner, C., Luijten, G., ... & Reyes, M. (2024). Medshapenet–a large-scale dataset of 3D medical shapes for computer vision. Biomedical Engineering/Biomedizinische Technik.
[2]. Teiga, I., Sousa, J. V., Silva, F., Pereira, T., & Oliveira, H. P. (2024, June). Radiological Medical Imaging Annotation and Visualization Tool. In International Conference on Human-Computer Interaction (pp. 317–333). Springer Nature Switzerland.
[3]. Kopowski, J., Mreła, A., Mikołajewski, D., & Rojek, I. (2024). Enhancing 3D Printing with Procedural Generation and STL Formatting Using Python. Applied Sciences, 14(16), 7299.
[4]. Bartol, I. R., Graffigna Palomba, M. S., Tano, M. E., & Dewji, S. A. (2024). Computational multiphysics modeling of radioactive aerosol deposition in diverse human respiratory tract geometries. Communications Engineering, 3(1), 152.
[5]. Badrou, A., Mariano, C. A., Ramirez, G. O., Shankel, M., Rebelo, N., & Eskandari, M. (2025). Towards constructing a generalized structural 3D breathing. PLOS Computational Biology, 21(1), e1012680.
[6]. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., & Xiao, J. (2015). 3D ShapeNets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1912–1920).
[7]. Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., ... & Savarese, S. (2015). ShapeNet: An information-rich 3D model repository. arXiv preprint arXiv:1512.03012.
[8]. Tulsiani, S., Zhou, T., Efros, A. A., & Malik, J. (2017). Multi-view supervision for single-view reconstruction via differentiable ray consistency. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2626–2634).
[9]. Tatarchenko, M., Dosovitskiy, A., & Brox, T. (2016). Multi-view 3D models from single images with a convolutional network. In European Conference on Computer Vision (pp. 322–337). Springer, Cham.
[10]. Yan, X., Yang, J., Yumer, E., Guo, Y., & Lee, H. (2016). Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In Advances in Neural Information Processing Systems (pp. 1696–17