A Close-Up on the AI Radiologist Software

Ayman Al-Kababji[1(✉)], Faycal Bensaali[1], and Sarada Prasad Dakua[2]

[1] College of Engineering, Qatar University, Doha, Qatar
{aa1405810,f.bensaali}@qu.edu.qa
[2] Department of Surgery, Hamad Medical Corporation, Doha, Qatar
SDakua@hamad.qa

Abstract. Proper computer-assisted detection (CADe) based on machine learning (ML) is a hot research topic in healthcare. Thousands of studies are published yearly focusing on enhancing the performance of ML-based models, but few tackle the challenge of deploying them efficiently. This paper focuses on designing an effective graphical user interface (GUI) tool, AI Radiologist, that clinicians can use during pre-operative planning to segment different liver tissues (parenchyma, tumors, and vessels). The tool employs convolutional neural networks (ConvNets) for liver tissue segmentation. This helps increase the success rate of any operation through meticulous pre-planning that allows surgeons to prepare well and plan for worst-case scenarios that might occur during surgery. AI Radiologist, an offline system application, utilizes three ConvNet models trained to segment all liver tissues. We use the PyQt5 Python module for the GUI to create a single-page application. The output of the AI Radiologist application is the liver, tumors, and vessels 2D slices and the 3D interpolation in .obj and .mtl format. The 3D interpolation can be visualized as a 3D liver object on any 3D-friendly software or 3D printed. Creating the AI Radiologist provides clinicians with a user-friendly GUI tool for liver tissues' segmentation and 3D interpolation, employing state-of-the-art models for all tissues' segmentation processes. Clinicians can select the volume(s) and the pre-trained models, and the AI Radiologist will take care of the rest.

Keywords: computer-assisted detection (CADe) systems · artificial intelligence · one-page application · liver tissues segmentation · graphical user interface (GUI)

1 Background

Two million tragic deaths worldwide are credited to hepatic-related diseases annually [5]. 50% of these deaths are caused by complications following liver

This publication was made possible by an Award [GSRA6-2-0521-19034] from Qatar National Research Fund (a member of Qatar Foundation). The contents herein are solely the responsibility of the authors. Moreover, the HPC resources and services used in this work were provided by the Research Computing group in Texas A&M University at Qatar. Research Computing is funded by the Qatar Foundation for Education, Science, and Community Development (http://www.qf.org.qa).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023, LNCS 13957, pp. 431–440, 2023.
https://doi.org/10.1007/978-3-031-36808-0_30

cirrhosis, while the other half are due to hepatitis and hepatocellular carcinoma (HCC) [5]. Moreover, it is a hub for metastasis originating from neighboring organs such as the colon, rectum, pancreas, stomach, esophagus, breasts, lungs, etc. [18]. Since tumors can originate from the liver or adjacent organs, the liver and its lesions are always analyzed in primary tumor staging [11], as the screening for liver-related diseases can reduce mortality [13]. Early detection and accurate delineation of hepatic tumors can dramatically help clinicians decide on more appropriate treatment plans.

Researchers usually offer interactivity within their applications (desktop/web) to give users the ability to control the application's flow or enhance the algorithms' segmentation. In [4], a desktop application, goes by the name CardiovasculaR Integrated Modelling and SimulatiON (CRIMSON), is created to provide a customizable, potent, and user-friendly system for performing 3D and reduced-order computational hemodynamic studies. Another well-known framework is the interactive learning and segmentation toolkit (ilastik) developed by [6]. It provides interactivity for application-domain experts to train classical machine learning (ML)-based biomedical image analysis tools for nuclei segmentation through their graphical user interface (GUI). This survey will not be complete without mentioning the famous ITK-SNAP application [21]. It is a classic for interactive segmentation and visualization tools for medical images.

In the case of [3], trainable WEKA segmentation (TWS), where WEKA stands for Waikato environment for knowledge analysis, is a pixel classification tool with a versatile pool of applications such as boundary detection, object detection and localization, and semantic segmentation for objects under a microscope.

On the other hand, the following studies have standalone applications, but they also deploy ML, ConvNets, and deep learning algorithms to tackle different challenges. In [12], an automated web-based application employing the famous U-Net as the deep learning model is utilized for spine segmentation. In [7], a web application named BreastScreening focuses on supplying AI-assisted techniques in the breast cancer classification challenge. In [16], FeAture Explorer (FAE) is a GUI created to allow researchers to develop classical supervised ML models, along with preprocessing techniques that are engineered to enhance their performance.

Some surveyed studies incorporate virtual reality (VR) technologies into the medical field for various applications. In [10], a collaborative VR environment is implemented, using Unity, Virtual Environment Toolkit (VRTK), and Photon frameworks, to assist surgeons in a remote or co-located environment in liver tumors surgery planning.

In the case of [9,20], the end goal is slightly different from the other surveyed literature in terms of the target audience. Chen et al. [9] create DeepLNAnno, which we assume stands for "Deep Lung Nodules Annotation" to discover and annotate lung nodules. It is deployed semi-automatically utilizing a simple 3D ConvNet, trained over adenocarcinoma and benign nodules, that annotates the nodules on unannotated CT images. Consequently, it produces reference images for human annotators, especially since they are Ph.D. candidates, not expert radiologists. The nature of the framework used to build the GUI is not disclosed.

In [20], RadCloud integrates the algorithmic aspect of ML, deep learning, and radiomics analysis functions over medical image processing with the data management one. The end-users for this platform are the researchers and scientists aiming to build better and enhanced tools to tackle different medical challenges. Their strategy is to host ML and deep learning applications and services on a cloud, providing low-cost, easy-to-use, and remotely-accessible data storage and sharing system. Some radiomics-related challenges investigate organs like the liver, breasts, kidneys, lungs, and rectum. They offer functionalities from automatic feature extraction, ML and ConvNets training, validation and testing, and hyperparameters tuning, providing scientists with an easy user experience (UX) for non-programmers.

One common theme that can be distinguished from all the surveyed studies is that they want to reduce the inconvenience that non-programmers and clinicians endure when creating/deploying statistical, ML, or deep learning models on various applications. This is exactly the aim of creating our artificially intelligent (AI) Radiologist tool for the liver tissues segmentation challenge. What we uniquely do in this paper is that we first built a tool to segment all the liver tissues from end to end in a sequential manner. Moreover, even though it is currently targeted for the liver, the software can be slightly modified to be used as is with any other organ where the organ needs to be segmented first, and the tumors and vessels within are also desired to be segmented. Lastly, we also avoid user interaction and do not presume significant pre-existing knowledge/experience to allow easier deployment to the extent that the user can figure out the mechanics of the application without any significant help.

This paper mainly focuses on developing the AI Radiologist system application but briefly mentions the implementation aspect of the deep learning technique. From this aspect, we highlight the significance of our work portrayed by Fig. 1. AI Radiologist allows clinicians to precisely segment the liver tissues (parenchyma, tumors, and vessels), followed by a 3D object creation using the segmented tissues. We create a highly intuitive tool such that the AI Radiologist system's learning curve is minimal and very efficient. The AI Radiologist system takes as an input a CT scan in "neuroimaging informatics technology initiative (NIfTI)" format, and the output is the generated masks for each tissue (NIfTI), along with the 3D liver object (.obj and .mtl). The reason for choosing the .obj and .mtl is because they can include multiple objects and are the file types used by 3D printers for multi-object printing. The main contributions presented in this paper can be summarized as follows:

- We present AI Radiologist, an AI-powered offline desktop application with pre-trained models for the liver segmentation tissues challenge. It offers extremely powerful computations but with minimal user interactivity and expertise.
- It extensively elaborates on the methodology of creating multi-tissued objects, which are conveniently the output of our AI Radiologist system, where they can be immediately 3D printed.

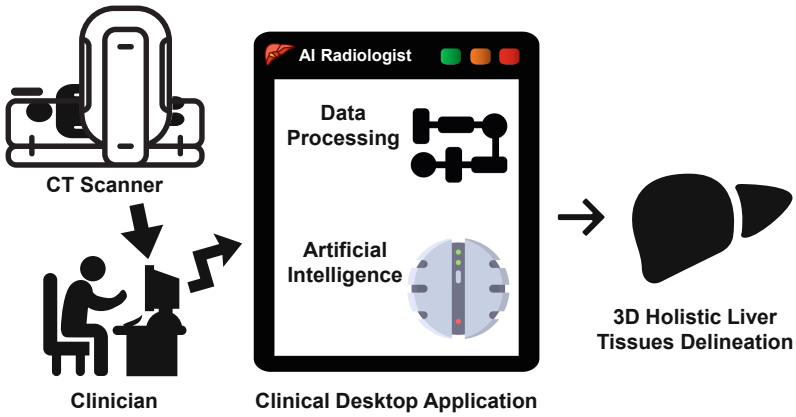

Fig. 1. AI Radiologist operation diagram.

The remainder of this paper is organized as follows. Section 2 describes the methodology followed to create the AI Radiologist application. Section 3 showcases the developed AI Radiologist application, and finally, we conclude the paper in Sect. 4.

2 Methodology

It is worth pointing out that the schema of our system to tackle the liver tissues segmentation always begins with the first ConvNet tackling the liver segmentation task. Then, its output is multiplied by the original CT slices to keep the liver's voxels and remove other voxels considered as background. Then, the CT slices, with liver voxels only, are fed into the tumors and vessels ConvNets in parallel to generate these tissues' masks. Lastly, the output of the three ConvNets is then combined and interpolated to create the 3D object of the liver, ready to be visualized or even printed.

2.1 Utilized Dataset and ConvNet

This paper aims to segment the liver parenchyma, tumors, and vessels; a complete dataset containing the manual segmentation for all three tissues is a must. The conducted review study in [2] shows that the dataset qualifying for this role is the medical segmentation decathlon challenge: task 8 hepatic vessel (MSDC-T8) dataset. It contains 443 contrast-enhanced (CE)-CT scans of varying voxels' dimensions between 0.58 and 0.97 mm for the intra-slice dimensions, while the inter-slice dimension varies between 0.8 and 8 mm. The number of slices varies between 24 and 251 slices. The liver ground-truth labels are provided in [17], while the tumors and vessels manual masks are provided by the challenge organizers in [15]. The liver masks are publicly available for the 443 records; however, the tumors and vessels masks are known for 303 only.

For the ML model, the U-Net ConvNet is utilized. It is worth mentioning that the size of the filters (f) is 3 (i.e., 3×3), with padding (p) being 1 and stride (s) being 1 are both applied to maintain the width and height of the convoluted layers so the outputs from the encoder side can be easily patched to the equivalent ones in the decoder side, without applying any cropping.

2.2 AI Radiologist Desktop Application

This subsection delves into the GUI design of the desktop application's schema and the framework used to create it. The application is designed in mind that it must be easy to use, intuitive, and with a high-quality UX.

Figure 2 shows the preliminary design of the AI Radiologist application. Users should be able to select the records they intend to segment by pressing the "Search Records" button. Once pressed, a pop-up window appears, allowing them to select multiple records to segment sequentially. Once the records have been selected, a table that shows the CT records' metadata is shown. The clinicians can examine the different information, such as the voxels' dimensions and the number of slices; the displayed metadata can be customized further based on the clinicians' needs. Following that, the ConvNets, that were trained on segmenting the liver's tissues can be selected by pressing the "Select ConvNets" button, which will pop up another window. Once selected, users can segment the tissues consecutively by pressing the "Segment Records". The achieved progress is displayed via a progress bar within the application and a status text message to allow for a better UX.

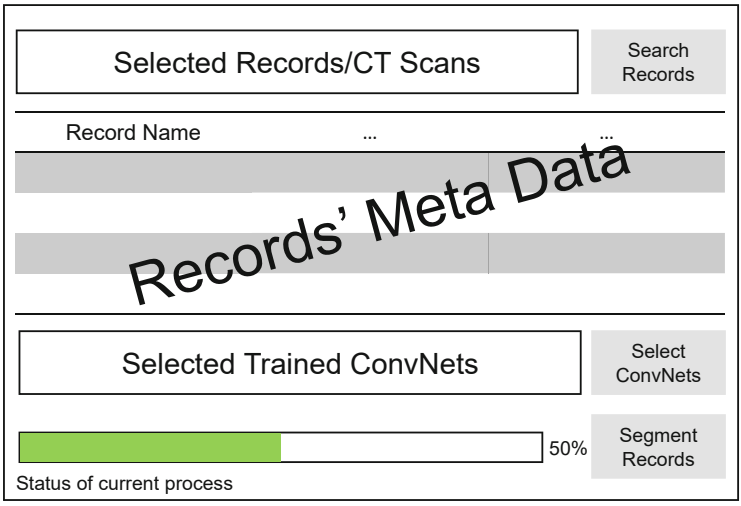

Fig. 2. Preliminary design of the AI Radiologist desktop application.

2.3 3D Construction and Printing

After creating the three ConvNets designed to segment the parenchyma, tumors, and vessels, creating a 3D object model of the liver's essential tissues has become attainable. We use "The Marching Cubes" algorithm [19] to interpolate the 2D cross-sectional slices into their 3D counterpart. It is deemed suitable as our focus is on the AI Radiologist system part implemented on the CT scans. However, we group the objects in a single .obj and .mtl file on Python from scratch.

Firstly, the three tissues' masks are loaded into the script after being generated by the designated ConvNets, via the nibabel Python library. Then, the 3D Lewiner interpolation using "The Marching Cubes" algorithm is used to create the 3D object of each tissue (the .obj file). It is worth noting that four mandatory types of parameters are mentioned in the .obj file, namely, the vertex (v), the vertex texture (vt), the vertex normal (vn), and the face (f).

The (v)s define the coordinates of vertices in the XYZ 3D space. On the other hand, the (vt)s and (vn)s are not as important, but the (vt)s aid in making the 3D object more realistic by mapping a pre-defined texture map on the faces, and the (vn)s aid in specifying the direction of the normal vector [8]. Lastly, the (f)s combine the three above components in the following order as shown by Eq. (1):

$$f \quad v_1[/vt_1][/vn_1] \quad v_2[/vt_2][/vn_2] \quad v_3[/vt_3][/vn_3] \quad \cdots \quad v_x[/vt_x][/vn_x] \quad (1)$$

where x is the number of vertices designated for each face f, as it can have as many vertices as needed.

The square brackets [ ] in Eq. (1) indicate that these values are optional. Moreover, because the scikit-image library generates vertices for triangular faces, three vertices are needed to build each face. Once the .obj file has been created for each tissue, the statement "mtllib <material_filename>.mtl" is added, enabling linkage with the .mtl file. The linking allows importing the defined texture and color onto different surfaces. Moreover, the statement "usemtl <tissue_material>" precedes the faces of an object in that file, allowing any software to assign the material color/texture to that object specifically. Lastly, .obj files for all tissues are then combined by placing all the (v)s of each .obj at the beginning, followed by all the (vt)s beneath the (v)s, and followed by the (vn)s beneath both (v)s and (vt)s. The same is applied to the (f)s; however, because more (v)s, (vt)s, and (vn)s have been added within the same file, the $2^{nd}, 3^{rd}, 4^{th}, \cdots, x^{th}$ vertices for the (f)s have to account for the number of vertices that are reserved for the preceding objects.

3 AI Radiologist Implementation and Showcase

We mention the trained ConvNets here in this paper to provide readers with context regarding what was used; however, more details about the development of models can be found in [1]. The focus is their employment within the AI Radiologist described within the following subsection.

This section outlines the next steps after creating the three ConvNet models developed to segment the three tissues (liver parenchyma, tumors, and vessels). We specifically use the PyQt5 framework to build the desktop application [14], available in Python. Figure 3 shows the main display that end-users get prompted with when they launch the application. The application is called "AI Radiologist" and has a liver icon for a logo.

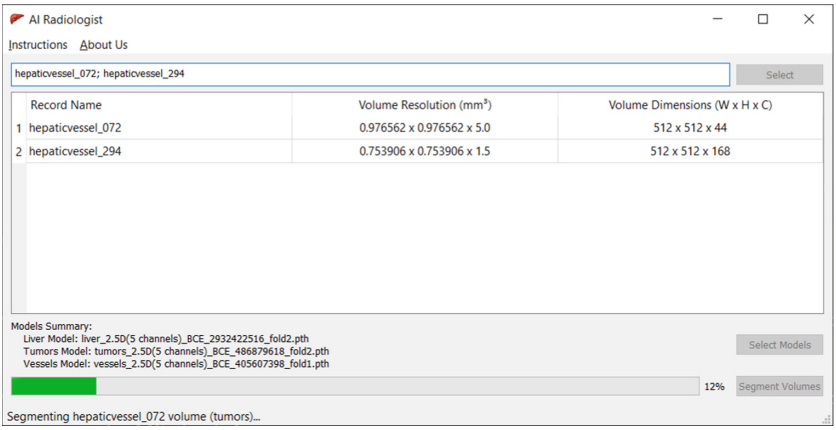

Fig. 3. Showing the progress in volumes processing.

When end-users press the "Select" button, a prompt pop-up window appears, enabling them to select multiple CT scans (saved as NIfTI format). Once CT scans are selected, a table shows the records' metadata for the end-user to observe. Moreover, the user must choose the PyTorch models trained to segment the liver, tumors, and vessels. Once selected, the application is ready to perform the segmentation.

Once the "Segment Volumes" button is pressed, the heavy-backend processes commence. Volumes undergo the same preprocessing techniques applied during the models' training phase. They are all done in real-time and efficiently, especially when using graphical processing units (GPU). We avoid using the main thread for heavy computations since it is responsible for all the GUI interactions. Hence, they are delegated to other subthreads. Figure 3 shows the effect of selecting multiple volumes for delineation and the chosen models to perform the segmentation process. Moreover, it shows the progress made by the application via the progress bar and the status bar.

The liver.nii.gz, tumors.nii.gz, and vessels.nii.gz have the segmented tissues in separate files for easier access, such that any medical software can open these masks, such as the ITK-SNAP software. Following that, the 3D interpolation process for the three generated masks commences. The resulting masks and the 3D object are placed in a folder holding the same name as the segmented volume within the same directory.

Moreover, the "complete_model.obj" file is created with all the vertices and faces needed to visualize the volume in 3D form. Lastly, the .mtl file adds color to different tissues to easily differentiate between them. Exporting both the .obj file and the .mtl file is the best option for multi-colored 3D printing [8]. The 3D interpolated liver volumes can be immediately put into a 3D printer; however, in the absence of such 3D printers, the 3D object can still be visualized using any 3D compatible software as seen in Fig. 4.

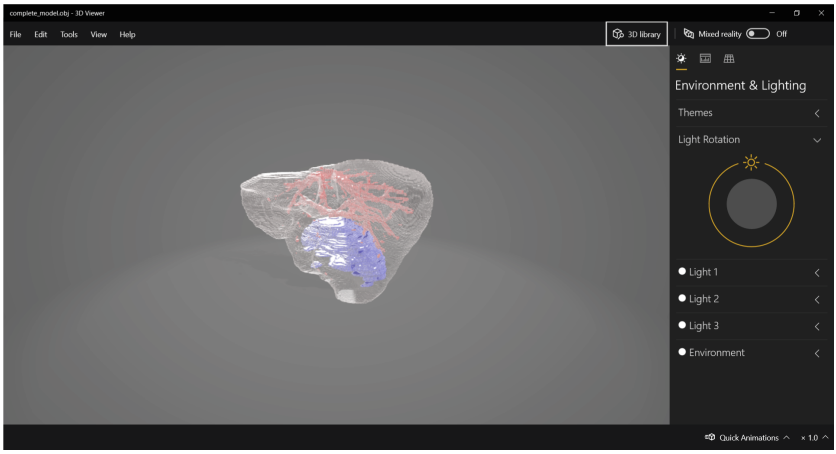

Fig. 4. Opening 3D interpolated object on 3D Paint software in Windows 10.

4 Conclusions and Future Directions

To conclude, by creating the AI Radiologist, we provide clinicians with an end-to-end user-friendly GUI tool for liver tissues segmentation that enables them to segment and 3D interpolate multiple records sequentially for medical analysis, with minimal possible human intervention. We are also employing state-of-the-art models for all tissue segmentation processes. All they need to do is select the volume(s), and select the models, and they will have to wait approximately a minute or two for each volume to be completed. It is worth re-iterating that volumes are done sequentially, meaning that the doctors can look into the first processed volume while waiting for the remaining ones.

In the future, we aim to enhance the AI Radiologist by adding the following features: i) Make it compatible with files that are of DICOM extension similar to the support we have for NIfTI files format; ii) Further enhance the segmentation results for both tumors and vessels tissues; and iii) Further develop the AI Radiologist to segment other tissues within the body such as lungs, pancreas, heart, and others. However, this requires available ConvNets trained over those tissues as well.

References

1. Al-Kababji, A., Bensaali, F., Dakua, S.P.: Scheduling techniques for liver segmentation: ReduceLROnPlateau vs OneCycleLR. In: Bennour, A., Ensari, T., Kessentini, Y., Eom, S. (eds.) Intelligent Systems and Pattern Recognition, pp. 204–212. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08277-1_17
2. Al-Kababji, A., Bensaali, F., Dakua, S.P., Himeur, Y.: Automated liver tissues delineation techniques: a systematic survey on machine learning current trends and future orientations. Eng. Appl. Artif. Intell. **117**, 105532 (2023)
3. Arganda-Carreras, I., et al.: Trainable weka segmentation: a machine learning tool for microscopy pixel classification. Bioinformatics **33**(15), 2424–2426 (2017). https://doi.org/10.1093/bioinformatics/btx180
4. Arthurs, C.J., et al.: CRIMSON: an open-source software framework for cardiovascular integrated modelling and simulation. PLoS Comput. Biol. **17**(5), 1–21 (2021). https://doi.org/10.1371/journal.pcbi.1008881
5. Asrani, S.K., Devarbhavi, H., Eaton, J., Kamath, P.S.: Burden of liver diseases in the world. J. Hepatol. **70**(1), 151–171 (2019). https://doi.org/10.1016/j.jhep.2018.09.014
6. Berg, S., et al.: Ilastik: interactive machine learning for (bio)image analysis. Nat. Methods **16**(12), 1226–1232 (2019). https://doi.org/10.1038/s41592-019-0582-9
7. Calisto, F.M., Santiago, C., Nunes, N., Nascimento, J.C.: Introduction of human-centric AI assistant to aid radiologists for multimodal breast image classification. Int. J. Hum. Comput. Stud. **150**, 102607 (2021). https://doi.org/10.1016/j.ijhcs.2021.102607
8. Chakravorty, D.: OBJ File Format - Simply Explained — All3DP (2021). https://all3dp.com/1/obj-file-format-3d-printing-cad/
9. Chen, S., Guo, J., Wang, C., Xu, X., Yi, Z., Li, W.: DeepLNAnno: a web-based lung nodules annotating system for CT images. J. Med. Syst. **43**(7), 1–9 (2019). https://doi.org/10.1007/s10916-019-1258-9
10. Chheang, V., et al.: A collaborative virtual reality environment for liver surgery planning. Comput. Graph. **99**, 234–246 (2021). https://doi.org/10.1016/j.cag.2021.07.009
11. Christ, P.F., et al.: Automatic Liver and Tumor Segmentation of CT and MRI Volumes using Cascaded Fully Convolutional Neural Networks (2017)
12. Kim, Y.J., Ganbold, B., Kim, K.G.: Web-based spine segmentation using deep learning in computed tomography images. Healthc. Inf. Res. **26**(1), 61–67 (2020). https://doi.org/10.4258/hir.2020.26.1.61
13. Nasiri, N., Foruzan, A.H., Chen, Y.W.: Integration of a knowledge-based constraint into generative models with applications in semi-automatic segmentation of liver tumors. Biomed. Signal Process. Control **57**, 101725 (2020). https://doi.org/10.1016/j.bspc.2019.101725. https://www.sciencedirect.com/science/article/pii/S1746809419303064
14. Phil Thompson: PyQt5 PyPI (2021). https://pypi.org/project/PyQt5/
15. Simpson, A.L., et al.: A large annotated medical image dataset for the development and evaluation of segmentation algorithms. arXiv preprint arXiv:1902.09063 (2019)
16. Song, Y., et al.: FeAture Explorer (FAE): a tool for developing and comparing radiomics models. PLoS ONE **15**, 1–10 (2020). https://doi.org/10.1371/journal.pone.0237587

17. Tian, J., Liu, L., Shi, Z., Xu, F.: Automatic couinaud segmentation from CT volumes on liver using GLC-UNet. In: Suk, H.-I., Liu, M., Yan, P., Lian, C. (eds.) MLMI 2019. LNCS, vol. 11861, pp. 274–282. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32692-0_32
18. UCSF Department of Surgery: Liver Metastases (2020). https://surgery.ucsf.edu/conditions-procedures/liver-metastases.aspx
19. van der Walt, S., et al.: scikit-image: image processing in Python. PeerJ **2**, e453 (2014). https://doi.org/10.7717/peerj.453
20. Yayuan, G., et al.: RadCloud-an artificial intelligence-based research platform integrating machine learning-based radiomics, deep learning, and data management. J. Artif. Intell. Med. Sci. **2**(1–2), 97 (2021). https://doi.org/10.2991/jaims.d.210617.001
21. Yushkevich, P.A., et al.: User-guided segmentation of multi-modality medical imaging datasets with ITK-SNAP. Neuroinformatics **17**(1), 83–102 (2018). https://doi.org/10.1007/s12021-018-9385-x