EVOLUTION IN ELECTRICAL AND ELECTRONIC ENGINEERING
e-ISSN: 2756-8458

EEEE

Vol. 5 No. 1 (2024) 69-76
https://publisher.uthm.edu.my/periodicals/index.php/eeee

Automatic 3D Segmentation of Liver Blood Vessels Using Deep Learning

Kumaran Naidu Subramaniam[1], Ashok Vajravelu[1]*, Asmarashid Ponniran[1]

[1] *Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Johor, 86400 MALAYSIA*

*Corresponding Author: ashok@uthm.edu.my
DOI: https://doi.org/10.30880/eeee.2024.05.01.010

Article Info

Received: 11 January 2024
Accepted: 21 February 2024
Available online: 30 April 2024

Keywords

IoT-based system, air quality, health complications, harmful gases, threshold value

Abstract

This study addresses the time-consuming and delicate nature of manually and automatically dividing hepatic blood vessels. The objective is to develop a rapid, highly accurate, and efficient autonomous division using deep learning. The proposed DeepVesselNet-FCN architecture involves data collection, pre-processing, cross-filter creation, extreme class balancing, and network execution. This model achieves a Dice coefficient of 0.8549, demonstrating good accuracy. This study concludes by offering a practical application, a simple program enabling physicians to automatically and accurately divide liver blood vessels in three dimensions.

1. Introduction

Generally, specialists use the medical technique of angiography to visualize the inside of the blood vessels and organs of the body. Angiographic information that has a 3d volume can be obtained using Magnetic Resonance Angiogram (MRA), ultrasound or x-ray. However, the interpretation of raw angiographic images to segment blood vessels does not provide sufficient information for clinical use, and other vessel characteristics such as centerline, diameter or vessel bifurcation are also required to extract information accurately about the vascular tree [1]. Therefore, a 3-Dimensional Convolutional Neural Network (CNN) can be used to Perform the task of vessel segmentation, centerline prediction and bifurcation detection. The 3D CNN is chosen so that the 3-D context information important for tracking the curved structure in 3-D is preserved. To reduce the high computational cost of 3D CNN and improve accuracy, cross-filtering and extreme class balancing with stable weights has been used [2].

Processing 3-D medical volumes is time consuming and memory intensive. When compared to a 2-D CNN, using a 3-D CNN results in a significant increase in the number of parameters to be optimized and calculations to be performed. Using a 2-D CNN in a slice approach, on the other hand, discards the critical 3-D context information needed to follow curvilinear structures in 3-D. The use of a cross-hair filter consisting of three intersecting 2-D filters is demonstrated, aiming to circumvent the memory and speed issues associated with classical 3-D networks. Additionally, 3-D information in volumetric data is incorporated through this approach [3][4]. Unlike previous approaches, which extract 2-D planes at a pre-processing step and use them as input channels, cross-hair filters are implemented at the layer level, allowing 3-D information to be preserved throughout the network.

High-class imbalances characterize vessel, midline and bifurcation prediction tasks. Vessels make up 3% of the total voxels in patient voxels, midlines make up a small percentage of segmented vessels, and the number of visible bifurcations is in the hundreds at most—even in voxels with 106 or more voxels. In medical data, bias against this background class is a common issue [5]. Unfortunately, in severe conditions like ours, the current class balancing loss function for CNN training turns out to be numerically unstable. The "severe class imbalance" problem is addressed by introducing a new loss function that is demonstrated to be effective with the interesting vascular properties.

2. Methodology

First the dataset has been obtained from IRCAD. In this process, publicly available liver CT scan datasets from Research Institute against Digestive Cancer (IRCAD) has been obtained and the CT scans are in 3D[6].Next is data interpolation. The CT scans are in different dimensions. In this process the data is interpolated to transform the dataset into similar voxel dimensions of $1x1x1mm^3$.In this step, the liver is separated from other organs in the CT scan, such as the spleen and stomach, as the CT scan covers a wide area of the hip section. Afterwards is the data augmentation. The existing data is too limited to train the deep learning model. The X and Y dimensions of each 2D sliced image of the CT scan are slightly changed to increase the dataset. The next step is data synthesization. In this step, the image data is converted into a 3D numpy array to be fed into the deep learning model because the original CT scan image data is not understandable by the deep learning model. Next is training the model. In this step, the model is trained using 80% of the dataset, and the remaining 20% of the dataset is utilized to test the trained model. Finally, adding the model to the interface. In this step, an interface is created using Python's tkinter library to enable the use of the DeepVesselNet-FCN model by anyone. Fig. 1 shows an overview of the flowchart of the system.

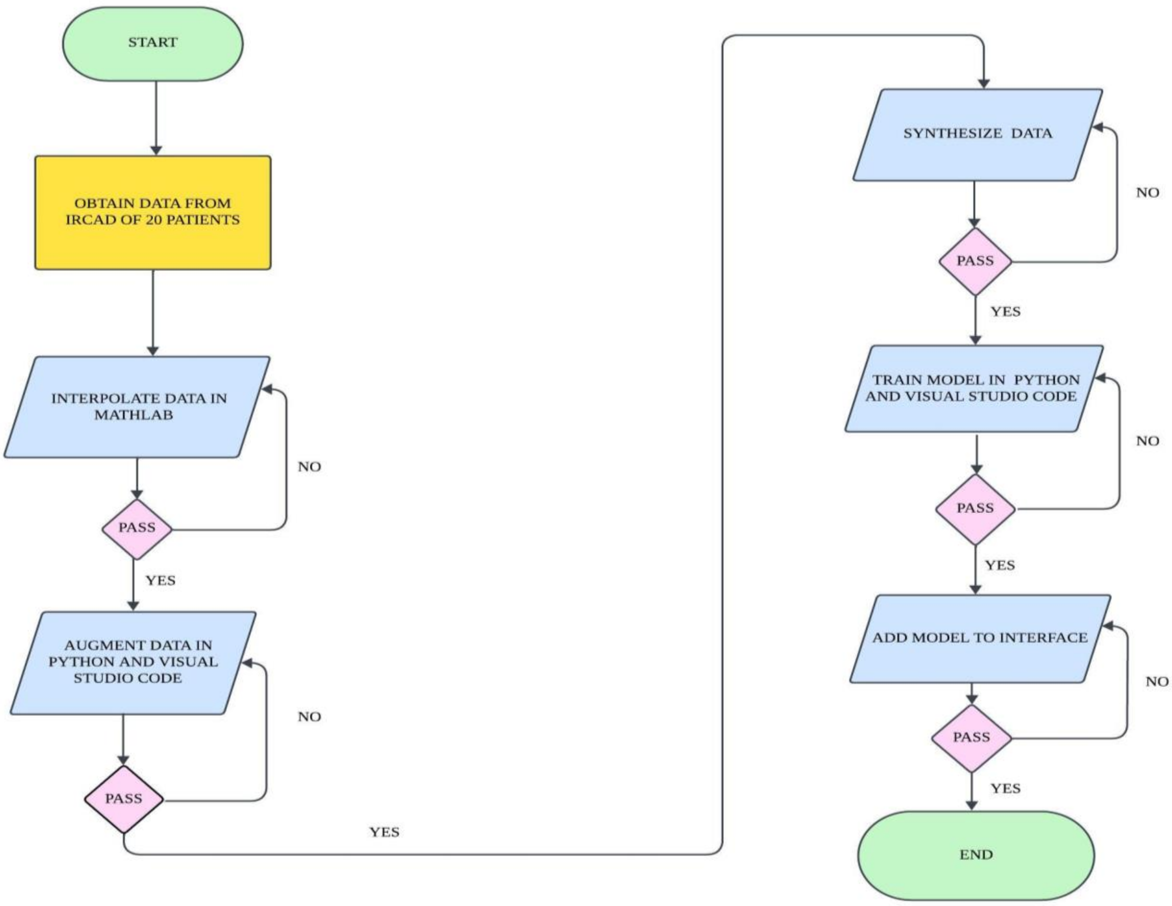


Fig. 1 *Flow chart diagram*

3. Results and Discussion

The findings of this study using DeepVesselNet-FCN have been recorded in this chapter. Accuracy is measured using the Dice coefficient. The Dice coefficient is a statistic used to measure the similarity of two samples. Or in this test is the similarity between the segmentation image processed by DeepVesselNet and the real segmentation image or known as "ground truth". Image segmentation accuracy in terms of DICE coefficients can be found in this chapter. Additionally, the original image and segmentation are also included below. Two alternative convolution-deconvolution architectures, 3D-UNet and VNet, were implemented to compare the performance of DeepVesselNet-FCN [7] [8]. This study was conducted using a system with 8GB of RAM and a 4GB Nvidia GeForce GTX 1650 Ti GPU. DeepVesselNet was trained using 100 original images and real liver segmentation. The data below is the test result of 10 heart liver samples.

3.1 3D-UNet

From Table 1, we can observe that using 3D-UNet without cross-filter and extreme class balancing with stable weights yields an acceptable dice coefficient of 0.7971. Notably, there are some tests that score over 0.8 dice coefficient with the highest being 0.8399. The lowest dice coefficient is 0.7700. Thus, based on these limited tests, the range percentage is 8.32%.

Table 1 *Output of 3D-UNet without cross-filter and extreme class balancing with stable weights*

Liver Verification Data	Number of actual Segmentation Pixels	Number of Output Segmentation Pixels	Dice Coefficient
1	224220	3181590	0.7701
2	150531	516633	0.8350
3	224220	3161706	0.7703
4	231675	3280119	0.7700
5	224220	3161706	0.7703
6	229311	3245793	0.7708
7	218265	3088956	0.7701
8	148737	482247	0.8398
9	157671	511179	0.8399
10	150531	516633	0.8350
Average	195938.1	2114656.2	0.7971

3.2 VNet

Based on Table 2, using VNet output without cross filter and extreme class balancing with stable weights yields a meager 0.4645 dice coefficient. The highest score is 0.4708 and lowest is 0.4621. Thus, the range percentage is 1.85%. The average dice coefficient is too low to make VNet a suitable deep learning model in this scenario.

Table 2 *VNet output without cross filter and extreme class balancing with stable weights*

Liver Verification Data	Number of actual Segmentation Pixels	Number of Output Segmentation Pixels	Dice Coefficient
1	224220	39991872	0.4640
2	150531	49405026	0.4622
3	224220	39727845	0.4642
4	231675	40391001	0.4708
5	224220	39727845	0.4642
6	229311	39768759	0.4683
7	218265	38671950	0.4647
8	148737	48730566	0.4625
9	157671	51577791	0.4621
10	150531	49405026	0.4622
Average	195938.1	43739768.1	0.4645

3.3 Model without initial training of DeepVesselNet-FCN

From Table 3, the average dice coefficient from using DeepVesselNet-FCN without initial training without cross filter and extreme class balancing with stable weights is a commendable 0.8383. Importantly, in some tests, the deep learning model is able to score significantly higher with the highest dice coefficient being 0.8899. The lowest dice coefficient is 0.8053. Hence, the range percentage is 9.51%. Although the fluctuating delta is higher, the dice coefficient is close and in certain tests significantly higher than the other deep learning models.

Table 3 *Output of DeepVesselNet-FCN without initial training without cross filter and extreme class balancing with stable weights*

Liver Verification Data	Number of Actual Segment Pixels	Number of Pixels Output Segment	Dice Coefficient
1	224220	1609698	0.8053
2	150531	121089	0.8899
3	224220	1571913	0.8059
4	231675	1622502	0.8053
5	224220	1571913	0.8059
6	229311	1607913	0.8057
7	218265	1530072	0.8058
8	148737	120231	0.8856
9	157671	127617	0.8839
10	150531	121089	0.8899
Average			0.8383

3.4 Model with initial training of DeepVesselNet-FCN

Based off Table 4, using a pretrained DeepVesselNet-FCN models is slightly better than the non-pretrained model with an average dice coefficient of 0.8549. The highest score is 0.8677 and the lowest is 0.8461. This gives a range percentage of 2.16%. Thus, using the pretrained model is able to give outputs with higher dice coefficients more consistently and is comparatively the best deep learning model.

Table 4 *Output of DeepVesselNet-FCN with initial training without cross filter and extreme class balancing with stable weights*

Liver Verification Data	Number of Actual Segment Pixels	Number Of Pixels of Output Segment	Dice Coefficient
1	224220	858594	0.8468
2	150531	39669	0.8664
3	224220	844851	0.8477
4	231675	867594	0.8461
5	224220	844851	0.8477
6	229311	858819	0.8470
7	218265	823761	0.8489
8	148737	39486	0.8677
9	157671	42312	0.8648
10	150531	39669	0.8664
Average			0. 8549

3.5 Comparison of time taken to train the deep learning model

The time taken for the DeepVesselNet-FCN model to complete training is also relatively short at 33 minutes. CNN training in 3D usually consumes a lot of time due to the processing power required to process the volumetric data. However, DeepVesselNet-FCN shows a slightly lower time to train. The DeepVesselNet-FCN model performs 27.27% and 51.52% faster than the 3D-UNet and VNet models, respectively. However, it is important to note that this performance may vary depending on system capabilities, number of epochs, number of input data and data size. In comparison, ("Study: 3D U-Net+ResNet — Volumetric Convolution + Long & Short Waste Connection (Biomedical Image Segment) | by SikHo Tsang | Towards Data Science" n.d.) took 4 hours to train a 3D deep learning model using 50 images at 320x320x60 dimensions with NVIDIA TITAN X GPU. Table 5 displays the comparison between the time taken to train deep learning models in minutes. The longest time taken is by VNet at 50 minutes and the shortest is DeepVesselNet-FCN. Both the pretrained and non-pretrained DeepVesselNet-FCN models took the same amount of time due to similar architecture. These results demonstrate that DeepVesselNet-FCN is the best deep learning model due to the shortest time taken to train it.

Table 5 *Time comparison between deep learning models*

Deep Learning Model	Time (min)

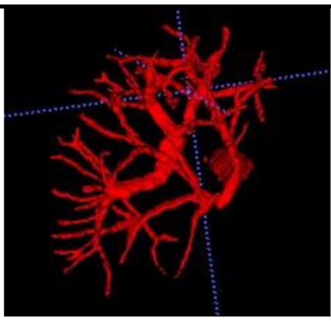
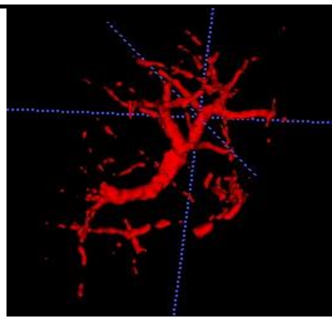
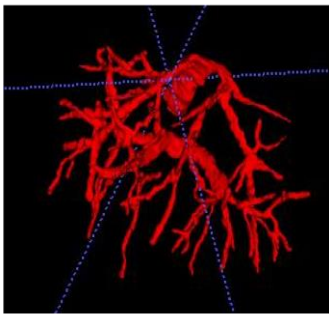
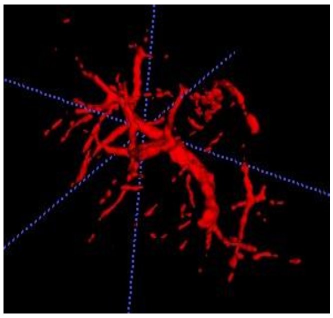
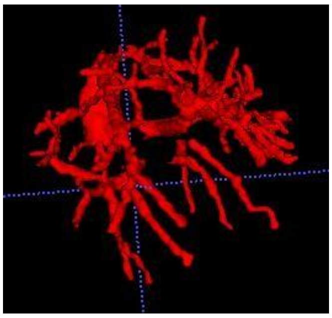
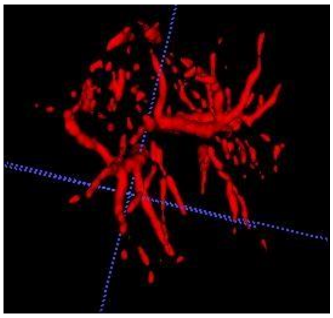
Evolution in Electrical and Electronic Engineering Vol. 5 No. 1 (2024) p. 69-76

3D-UNet	42
VNet	50
DeepVesselNet-FCN without Initial training	33
DeepVesselNet-FCN with initial training	33

3.6 Comparison in three dimensions(3D)

Based on Table 6 which encompasses the three-dimensional comparisons between actual segmentation and the output segmentation from DeepVesselNet-FCN pretrained model, our analysis of the two-dimensional comparison is further justified. Majority features of the blood vessels such as the bifurcations, directions and vessel diameters are present. However, some blood vessel branches are thinner or missing compared to the actual segmentation. The missing blood vessels can be attributed to the output segmentation of the vessel being too small to be detected as a bifurcation or branching vessel. With further training on a more diverse dataset, the deep learning model should improve in terms of the diameter of the segmented blood vessel with will directly corelate to a reduction in loss of bifurcation detection. However, despite some minor blood vessels being thinner or missing bifurcations, the current accuracy of the deep learning model is satisfactory as it is able to segment the most of the major blood vessels with high accuracy. For future work, the features of minor blood vessels such as smaller bifurcations and diameter should be improved on. These improvements can be made by training on a specific dataset of smaller blood vessels to improve the deep learning model's detection on its features.

Table 6 *3D comparison of liver blood vessels*

Liver Verification Data	Real Segmentation Image	Output Segmentation Image
		
		
		

3.7 Graphical User Interface (GUI)

Finally, it can be observed that the graphical user interface (GUI) allows the easy segmentation of liver blood vessels in 3D by anyone. Practical applications of research outputs are not often presented in most papers. A simple and modern GUI based on Python has been created using the tkinter module. Additionally, modules like pyinstaller have been utilized to make this GUI executable. Therefore, anyone can use this GUI to test or create a 3D segmentation of liver blood vessels without any coding. This will be very beneficial for medical professionals with little or no background in coding. Since this GUI can also use for testing, this interface can be used to test accuracy deep learning model easily in terms of Dice coefficient. After segmentation done, the user will be presented with all the important details i.e. the number output pixels, number of segment pixels and Dice coefficient. As an added benefit using python's pyinstaller module, this program can be run anywhere computer without having to install python or anything extra. Therefore, this program can be widely used anywhere on any computer without any additional requirements required.

3.7.1 GUI before segmentation begins

Fig. 2 shows the tkinter based GUI before segmentation is started. There are 2 required inputs, "Select Input Image" which selects the raw CT scan images as input and "Select Output Folder" which selects the folder the output segmentation should be saved in. There is also 1 optional input, "Select Input Label" which selects the ground truth of a segmentation in order to test and compare the dice coefficient to our DeepVesselNet-FCN pretrained model. Once all the required inputs are given, the "START SEGMENTATION" button can be pressed to start the segmentation. There are 4 empty outputs, "Status" which states the processing status, "No. Output Pixels" which states the segmented output pixels from out deep learning model, "No. Segment Pixels" which states the number of segment pixels from the ground truth and "Dice" which states the dice coefficient of our deep learning model's output.

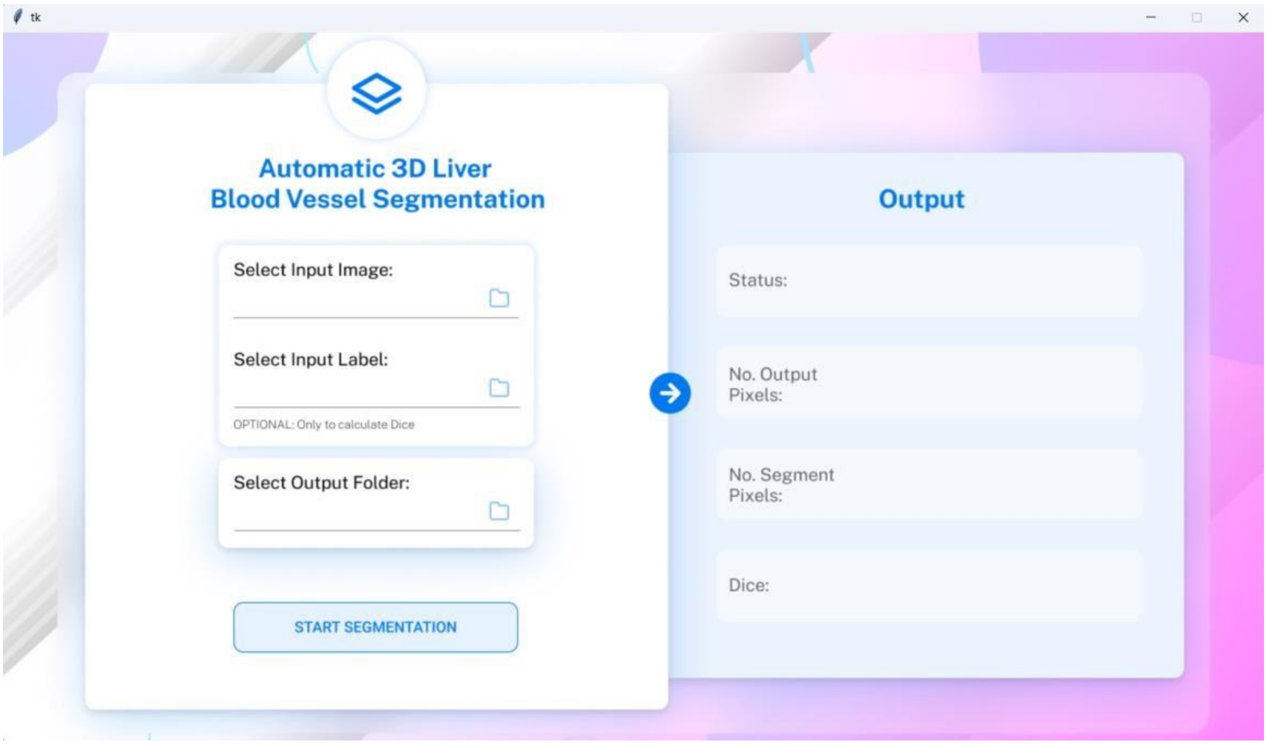


Fig. 2 *GUI before segmentation begins*

3.7.2 GUI after segmentation is complete

From Fig. 3 it shows the GUI after segmentation using DeepVesselNet-FCN pretrained model is completed. All the inputs have been selected and the file paths are displayed in the GUI to inform the user. On the right, all the output values have been updated. The "Status" shows "Segmentation Completed" to inform the user that the deep learning model have finished processing and saved the output segmentation. The "No. Output Pixels", "No. Segment Pixels" and "Dice" sections are populated accordingly to inform the user.

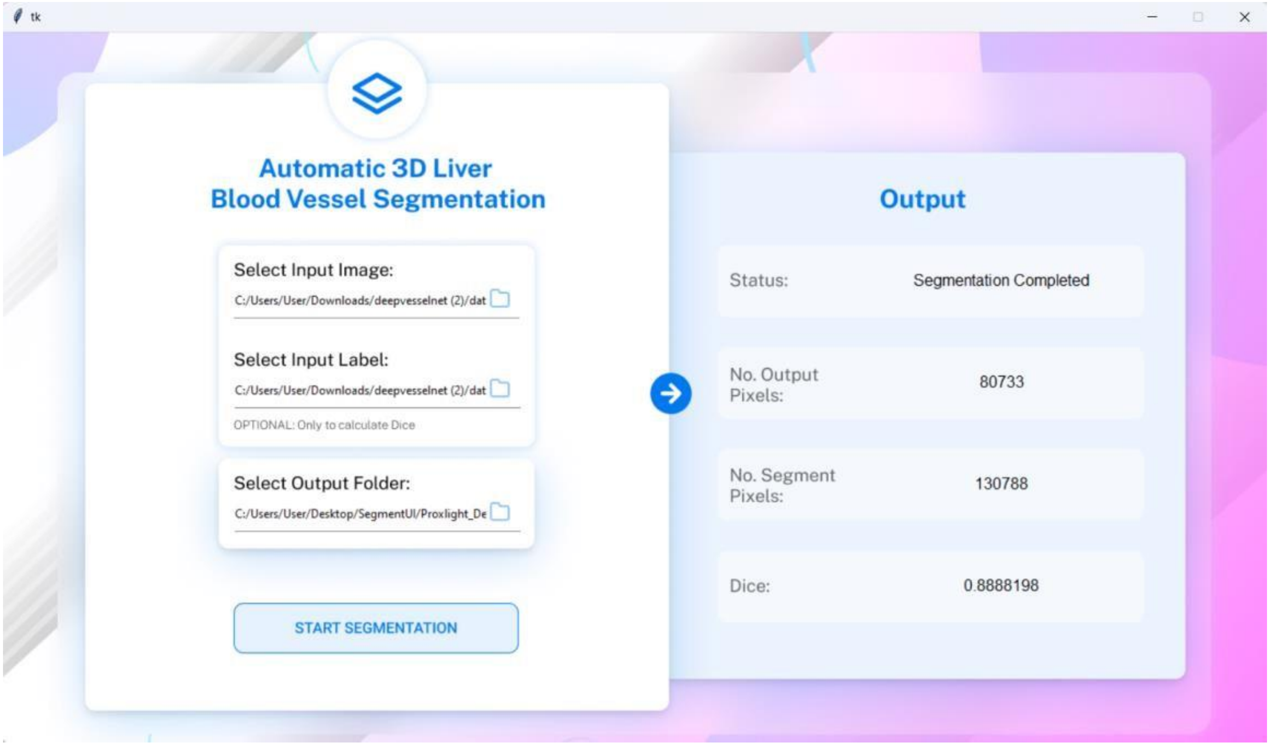


Fig. 3 *GUI after segmentation is complete*

3.8 Instructions to use the GUI

To download and use our GUI, begin by downloading it from the provided link: https://drive.google.com/file/d/1pX_EK40bDTxwY8v1JwwjMnZlZRm6oDE3/view?usp=drive_link.

Once downloaded, extract the zipped folder. If you are a Windows user, navigate to the GUI folder and double-click on the "GUI.exe" file to execute it. For Linux/Mac users, delete the "dist" folder and the "GUI.exe" shortcut from the GUI folder. Make sure Python and PyInstaller are installed on your system. Open a terminal at the GUI folder location and input the following command: pyinstaller --onedir -w GUI.py. After execution, you can remove the "build" folder and the "GUI.spec" file. Now, open the "dist" folder, then the "GUI" folder, and run the "GUI" executable file to launch the GUI program. Additionally, to obtain the dataset, download it from this link: https://drive.google.com/file/d/1LbRIgyyezgV_ZyXsImVz29ME-pKgwmQF/view?usp=drive_link. Once downloaded, extract the zipped folder.

4. Conclusion

Overall, this study found that, DeepVesselNet-FCN can used to segment liver blood vessels in 3D. This model is capable to produce good results. Moreover, the time taken to train the model also short. This model is able to segment raw images in a short period of time.Transfer learning results in a slight increase in accuracy in terms of Dice coefficient. This paper shows that DeepVesselNetFCN can be used for liver blood vessels with some modifications and preprocessing. GUI made in the paper gives this research a practical application for experts medicine and research. Anyone can easily use this pre-trained model to segment liver blood vessels in 3D Liver blood vessels was successfully segmented automatically from 3D CT images. Power requirements 3D computing for 3D CDN is reduced. The resulting blood vessel segment have a good accuracy. This research has succeeded in achieving all the objective.

Acknowledgement

The author would like to thank the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia for all the support given throughout this work. Communication of this research is made possible through monetary assistance by Universiti Tun Hussein Onn Malaysia and the UTHM Publisher's Office via Publication Fund E15216.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The author attests to having sole responsibility for the following: planning and designing the study, data collection, analysis and interpretation of the outcomes, and paper writing.

References

[1] Tetteh, G., Efremov, V., Forkert, N.D., Schneider, M., Kirschke, J., Weber, B., Zimmer, C., Piraud, M. & Menze,B.H. 2020. DeepVesselNet: Vessel Segmentation, Centerline Prediction, and Bifurcation Detection in 3-D Angiographic Volumes. Frontiers in Neuroscience 14: 1285.

[2] Zheng, Y., Liu, D., Georgescu, B., Nguyen, H. & Comaniciu, D. 2015. 3D Deep Learning for Efficient and Robust Landmark Detection in Volumetric Data. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9349: 565–572.

[3] Becker, C., Rigamonti, R., Lepetit, V. & Fua, P. 2013. Supervised Feature Learning for Curvilinear Structure Segmentation. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8149 LNCS(PART 1): 526–533.

[4] Liu, S., Zhang, D., Song, Y., Peng, H. & Cai, W. 2017. Triple-Crossing 2.5D Convolutional Neural Network for Detecting Neuronal Arbours in 3D Microscopic Images. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 10541 LNCS*: 185–193.

[5] Christ, P.F., Elshaer, M.E.A., Ettlinger, F., Tatavarty, S., Bickel, M., Bilic, P., Rempfler, M., Armbruster, M., Hofmann, F., D'Anastasi, M., Sommer, W.H., Ahmadi, S.A. & Menze, B.H. 2016. Automatic Liver and Lesion Segmentation in CT Using Cascaded Fully Convolutional Neural Networks and 3D Conditional Random Fields. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9901 LNCS: 415–423.

[6] Liver segmentation – 3D-ircadb-01. (n.d.). IRCAD. https://www.ircad.fr/research/data-sets/liver-segmentation-3d-ircadb-01/

[7] Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T. & Ronneberger, O. 2016. 3D U-net: Learning dense volumetric segmentation from sparse annotation. *Lecture Notes in Computer Science (Including Subseries* Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9901 LNCS: 424–432.

[8] Milletari, F., Navab, N. & Ahmadi, S.A. 2016. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016 565–571.