HHS Public Access

Author manuscript

Proc SPIE Int Soc Opt Eng. Author manuscript; available in PMC 2017 June 12.

Published in final edited form as:

Proc SPIE Int Soc Opt Eng. 2017 February 11; 10135: . doi:10.1117/12.2256011.

DeepInfer: Open-Source Deep Learning Deployment Toolkit for Image-Guided Therapy

Alireza Mehrtash[a,b], **Mehran Pesteie**[a], **Jorden Hetherington**[a], **Peter A. Behringer**[b], **Tina Kapur**[b], **William M. Wells III**[b], **Robert Rohling**[a,c], **Andriy Fedorov**[b], and **Purang Abolmaesumi**[a]

[a]Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada

[b]Department of Radiology, Brigham and Women's Hospital, Boston, MA, United States

[c]Department of Mechanical Engineering, University of British Columbia, Vancouver, BC, Canada

Abstract

Deep learning models have outperformed some of the previous state-of-the-art approaches in medical image analysis. Instead of using hand-engineered features, deep models attempt to automatically extract hierarchical representations at multiple levels of abstraction from the data. Therefore, deep models are usually considered to be more flexible and robust solutions for image analysis problems compared to conventional computer vision models. They have demonstrated significant improvements in computer-aided diagnosis and automatic medical image analysis applied to such tasks as image segmentation, classification and registration. However, deploying deep learning models often has a steep learning curve and requires detailed knowledge of various software packages. Thus, many deep models have not been integrated into the clinical research workflows causing a gap between the state-of-the-art machine learning in medical applications and evaluation in clinical research procedures. In this paper, we propose "DeepInfer" – an open-source toolkit for developing and deploying deep learning models within the 3D Slicer medical image analysis platform. Utilizing a repository of task-specific models, DeepInfer allows clinical researchers and biomedical engineers to deploy a trained model selected from the public registry, and apply it to new data without the need for software development or configuration. As two practical use cases, we demonstrate the application of DeepInfer in prostate segmentation for targeted MRI-guided biopsy and identification of the target plane in 3D ultrasound for spinal injections.

1. INTRODUCTION

Medical image analysis is a crucial step in image-guided therapy (IGT) that can assist with detection and localization of the desired target tissue and may lead to improved treatment outcomes. Therefore, the majority of IGT procedures require quick, accurate and reliable image analysis methods prior to, or during the procedure. Over the past few years, deep

Further author information: (Send correspondence to UBC Electrical and Computer Engineering. 112B - 2356 Main Mall, Vancouver BC V6T1Z4, Canada), Alireza Mehrtash: mehrtash@ece.ubc.ca.

neural networks have shown great success in medical image processing, and numerous deep models have been proposed for different tasks, such as segmentation,[1–3] classification and detection,[4–6] scoring[7] and registration.[8] Deep convolutional neural networks (CNNs) have outperformed the previous state-of-the-art approaches in tackling medical imaging challenges. For instance, Ronneberger et al.[9] proposed a deep convolutional network for biomedical image segmentation which was ranked first in the ISBI challenge for segmentation of neural structures in electron microscopic stacks as well as the cell tracking challenge in 2015. However, despite recent advances in developing highly accurate deep models, deploying these technologies in clinical practice needs integration of many software packages and components into the existing platforms, which are designed for visualization and computerized image analysis, such as 3D Slicer*.[10] This procedure often requires technical software knowledge and a detailed understanding of the functionality of each of the libraries. As a result, many current deep learning models have not been incorporated in IGT workflows and there is a gap between the cutting edge technology and medically trained researchers who are limited by the tools that are already available in image analysis software. Therefore, there is a need for a solution which can seamlessly integrate deep learning technology into the existing medical image analysis platforms. In this paper, we propose a flexible open-source toolkit, called "DeepInfer"†, which can deploy multi-platform trained deep learning models and integrates them into 3D Slicer. DeepInfer allows machine learning experts to package, ship, and deploy their models, thereby additionally allowing clinical researchers and biomedical engineers to use task-specific deep models, without the need for further software development and configuration. Integration with 3D Slicer enables the user to leverage its powerful and versatile visualization and analysis capabilities, and makes it possible to combine conventional image analysis tools with the deep learning approaches.

2. METHODS

In this section we describe the architecture of our proposed toolkit. We then discuss the components of the system, followed by the prediction pipeline.

2.1 Architecture

Figure 1 shows the architecture of the DeepInfer toolkit, which has three components: the Docker‡11 engine, the DeepInfer 3D Slicer extension, and the cloud model registry. The Docker engine consists of local Docker containers that include the deployed models, as well as all of the required deep learning frameworks, which process incoming data and produce respective results. The Docker images can benefit from the power of Graphic Processing Units (GPUs) if they are available on the local machine. It is also possible to use only CPU for tasks which do not have extensive computing requirements. Using a highly parallel computing architecture, GPUs considerably increase the processing performance, consequently decreasing computation time, which can be significant in an IGT setting. The DeepInfer module is implemented as a 3D Slicer software extension which is responsible for

*http://slicer.org
†http://deepinfer.org
‡https://docker.com

communicating with the Docker engine and cloud model registry. It also provides a GUI for the selection of input and output nodes, as well as for visualization, in the data processing pipeline. Data streaming between the Docker engine and the GUI is achieved by sharing a local folder with the specific Docker image. Both 3D Slicer software and the DeepInfer extension are publicly available under non-restrictive BSD-style license.

2.2 System Components

2.2.1 Docker Engine—We used the Docker ecosystem[11] for deployment of the trained models. Docker is a containerization platform, which can be used to deploy software that will run the same on any system, regardless of its environment. Docker wraps software into standardized units, which encapsulate all the dependency tools and libraries. These units are called “images”, which are built into immutable files. Once running, they are called “containers”. Docker Hub[§] serves as a public repository of Docker images, which can, but do not have to be open-source.

In DeepInfer, a Docker container with the specific machine learning model communicates with the 3D Slicer extension through system commands and shared working folders. The dependencies and frameworks of the specific models that will be deployed are included in the Docker image. Since the Docker image is generated from generic Linux images, it is possible to install any of the available machine learning systems, such as Caffe,[12] Theano[13] and TensorFlow.[14]

2.2.2 Cloud Components—The trained models are stored as Docker images in the Docker Hub, as well as in the form of JavaScript Object Notation (JSON) description files on the GitHub model registry. Users can browse through different available models on the Slicer DeepInfer module and download the models from cloud. Meta-information about the models is described in a standard JSON file that contains the description, version, training and validation accuracies and the license of the model. The JSON meta-data file also provides the summary and types of inputs and outputs for generation of the user interface and the processing command-line. For example, some models will output segmentation maps, while others might output classification results. The core of each model is the model-specific architecture and the learned weights for the deep neural network parameters, which are deployed and shipped through the Docker images.

2.2.3 Slicer Extension—The user interface is implemented as a 3D Slicer extension, which is developed as a scripted python module with a custom user interface to perform the following: connect to the model registry; download a requested model; generate the Qt GUI based on local model specifications; send data for processing; and receive respective results from the deployed model. It also provides a visualization pipeline for the results. The module can be downloaded and installed directly from 3D Slicer Extensions Manager.

[§]https://hub.docker.com/

2.3 Usage Pipeline

To use DeepInfer, a user must install Docker and 3D Slicer onto their machine, adding the DeepInfer extension to 3D Slicer from the 3D Slicer Extensions Manager. The module connects to the GitHub model registry and receives a list of available models on the cloud along with their descriptions through the GitHub representational state transfer (REST) application program interface (API). The GUI presents the available model descriptions to the user. If the model is not available on the local machine, the user can select the desired model for the particular application and download the Docker image from the Docker Hub to the local machine. To run a specific algorithm on data, the user selects a model from the available containers on the local machine. Once selected, the corresponding JSON meta-data file is parsed to create the required GUI components, such as model-specific input-parameter fields. Imaging and non-imaging data are written to a folder on the local machine that is shared between Docker and 3D Slicer. A Docker command is executed to load the model with its respective parameters, which can then communicate to DeepInfer and process input data. The output results is written back to the shared folder, where the DeepInfer extension will read the results and present them to Slicer for visualization.

3. RESULTS

Various IGT applications can benefit from the proposed toolkit. In this section we describe the initial use cases of DeepInfer toolkit in supporting interventional IGT procedures.

3.1 Prostate Segmentation in Targeted MRI-Guided Biopsy

There is growing evidence demonstrating superiority of MRI-guided targeted biopsy over the standard of care sextant approaches driven purely by ultrasound. Over the past years, we have been developing an approach to targeted biopsy where the patient is located inside the MRI scanner bore throughout the procedure, and the targeted samples are collected using the transperineal approach.[15] This approach relies on deformable image registration for re-identification of the targets defined on the planning scans during the procedure.[16,17] In this setting, robust automatic segmentation of the prostate gland can shorten the manual interaction time, and can contribute to the overall reduction of the procedure time.

We used the contours of the prostate collected during MR-guided prostate biopsy procedures to train a deep neural network based on a customized variant of the U-Net architecture.[9] The network was trained on N = 224 patients on a total number of 26250 2D slices of prostate images and achieved 76.25% accuracy on N = 57 validation patients (2184 2D slices). The model was deployed using our presented deployment toolkit and was successfully tested to perform an end-to-end prostate segmentation task. The server was equipped with an NVIDIA GeForce GTX 980 Ti (6GB on-board memory). Figure 2 shows the client side visualization GUI and the results of processing.

3.2 3D Ultrasound Spine Plane Classification

Spinal injections, including facet joint injection and epidural needle placement, are typically performed as a common treatment of low back pain and local anesthesia. Currently, spinal injections are guided based on either fluoroscopy and Computed Tomography (CT) or

manually via palpation and loss-of-resistance technique which is a blind procedure for epidural injections. Due to the significant drawbacks of CT and flouroscopy such as exposure to the ionizing radiation, their application is prohibited on pregnant patients for epidural anaesthesia during delivery. Moreover, the loss-of-resistance technique for epidural needle placement has a failure rate within 6–20%.[18]

Ultrasound, as a non-ionizing imaging modality, has been investigated for real-time visualization of the spinal anatomy prior to or during injection therapy.[19–22] However, because of the complex spinal anatomy and their similar wave-like patterns (see Figure 3), a key challenge in using ultrasound for needle guidance is interpreting the ultrasound images and accurately identifying the target plane. Therefore, a system which automatically provides the anesthesiologist with feedback for identification of the target plane would be beneficial. Previously, we developed a system for automatic ultrasound plane classification for spinal injections namely, epidural and facet joint injections.[23] The system was trained on 1090 planes of 3D ultrasound volumes which were collected from 13 volunteers. Annotations of an expert sonographer was used for training to learn the signatures of the anatomical landmarks of the target planes. The model classification accuracy was 94% and 90% of the epidural and facet joint target images, respectively in leave-one-out cross validation. We deployed the model in the proposed toolkit to perform plane classification for epidural injections. Figure 4 shows the Slicer user interface to visualize the classification results.

4. DISCUSSION AND CONCLUSIONS

We proposed an open-source toolkit called “DeepInfer” for deploying deep models for image-guided therapy applications in 3D Slicer. The toolkit consists of a 3D Slicer extension module as user interface and a Docker engine for deployment of the trained models and a cloud system for storing different models. DeepInfer enables clinical and biomedical researchers to utilize task-specific, trained models, without the need for further software configuration, thus bridging the gap between current state-of-the-art machine learning methods for biomedical image analysis and medical researchers. Moreover, the flexible architecture of DeepInfer allows the user to deploy a wide range of trained machine learning models.

Acknowledgments

Research reported in this publication was supported by NIH Grant No. P41EB015898.

References

1. Pereira S, Pinto A, Alves V, Silva CA. Brain tumor segmentation using convolutional neural networks in mri images. IEEE transactions on medical imaging. 2016; 35(5):1240–1251. [PubMed: 26960222]
2. Brosch T, Tang L, Yoo Y, Li D, Traboulsee A, Tam R. Deep 3D Convolutional Encoder Networks with Shortcuts for Multiscale Feature Integration Applied to Multiple Sclerosis Lesion Segmentation. IEEE Transactions on Medical Imaging. 2016; 0062(c):1–1.
3. Ghafoorian, M., Karssemeijer, N., Heskes, T., van Uden, I., de Leeuw, FE., Marchiori, E., van Ginneken, B., Platel, B. International Symposium on Biomedical Imaging (ISBI). IEEE; 2016. Non-

uniform patch sampling with deep convolutional neural networks for white matter hyperintensity segmentation; p. 1414-1417.

4. Dou Q, Member S, Chen H, Member S, Yu L, Zhao L, Qin J. Automatic Detection of Cerebral Microbleeds from MR Images via 3D Convolutional Neural Networks. 2016; 0062(c):1182–1195.
5. Anthimopoulos M, Christodoulidis S, Ebner L, Christe A, Mougiakakou S. Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network. IEEE Transactions on Medical Imaging. 2016; 0062(c):1207–1216.
6. Ghafoorian M, Karssemeijer N, Heskes T, Bergkamp M, Wissink J, Obels J, Keizer K, de Leeuw FE, van Ginneken B, Marchiori E, Platel B. Deep multi-scale location-aware 3d convolutional neural networks for automated detection of lacunes of presumed vascular origin. NeuroImage: Clinical. 2017
7. Petersen, K., Nielsen, M., Diao, P., Karssemeijer, N., Lillholm, M. International Workshop on Digital Mammography. Springer; 2014. Breast tissue segmentation and mammographic risk scoring using deep learning; p. 88-94.
8. Miao S, Wang ZJ, Liao R. A cnn regression approach for real-time 2d/3d registration. IEEE transactions on medical imaging. 2016; 35(5):1352–1363. [PubMed: 26829785]
9. Ronneberger, O., Fischer, P., Brox, T. International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer; 2015. U-net: Convolutional networks for biomedical image segmentation; p. 234-241.
10. Fedorov A, Beichel R, Kalpathy-Cramer J, Finet J, Fillion-Robin JC, Pujol S, Bauer C, Jennings D, Fennessy F, Sonka M, et al. 3d slicer as an image computing platform for the quantitative imaging network. Magnetic resonance imaging. 2012; 30(9):1323–1341. [PubMed: 22770690]
11. Merkel D. Docker: Lightweight linux containers for consistent development and deployment. Linux J. 2014 (Mar. 2014).
12. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093. 2014
13. Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints abs/1605.02688. May.2016
14. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467. 2016
15. Penzkofer T, Tuncali K, Fedorov A, Song SE, Tokuda J, Fennessy FM, Vangel MG, Kibel AS, Mulkern RV, Wells WM, Hata N, Tempany CMC. Transperineal in-bore 3-T MR imaging-guided prostate biopsy: a prospective clinical observational study. Radiology. Jan.2015 274:170–180. [PubMed: 25222067]
16. Fedorov A, Tuncali K, Fennessy FM, Tokuda J, Hata N, Wells WM, Kikinis R, Tempany CM. Image registration for targeted MRI-guided transperineal prostate biopsy. Journal of Magnetic Resonance Imaging. 2012; 36(4):987–992. [PubMed: 22645031]
17. Behringer PA, Herz C, Penzkofer T, Tuncali K, Tempany CM, Fedorov A. Open-Source Platform for Prostate Motion Tracking During in-Bore Targeted MRI-Guided Biopsy. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2016; 8680:122–129.
18. Le Coq G, Ducot B, Benhamou D. Risk factors of inadequate pain relief during epidural analgesia for labour and delivery. Canadian Journal of Anaesthesia. 1998; 45(8):719–723. [PubMed: 9793659]
19. Karmakar M, Li X, Ho AH, Kwok W, Chui P. Real-time ultrasound-guided paramedian epidural access: evaluation of a novel in-plane technique. British Journal of Anaesthesia. 2009; 102(6):845–854. [PubMed: 19398454]
20. Lee, A. Seminars in Perinatology. Vol. 38. Elsevier; 2014. Ultrasound in obstetric anesthesia; p. 349-358.
21. Lin TL, Chung CT, Lan HHC, Sheen HM. Ultrasound-guided facet joint injection to treat a spinal cyst. Journal of the Chinese Medical Association. 2014; 77(4):213–216. [PubMed: 24631041]

22. Santiago AEQ, Leal PC, Bezerra EHM, Giraldes ALA, Ferraro LC, Rezende AH, Sakata RK. Ultrasound-guided facet block to low back pain: a case report. Brazilian Journal of Anesthesiology (English Edition). 2014

23. Pesteie M, Abolmaesumi P, Ashab HAD, Lessoway VA, Massey S, Gunka V, Rohling RN. Real-time ultrasound image classification for spine anesthesia using local directional hadamard features. International Journal of Computer Assisted Radiology and Surgery. 2015; 10(6):901–912. [PubMed: 26026697]

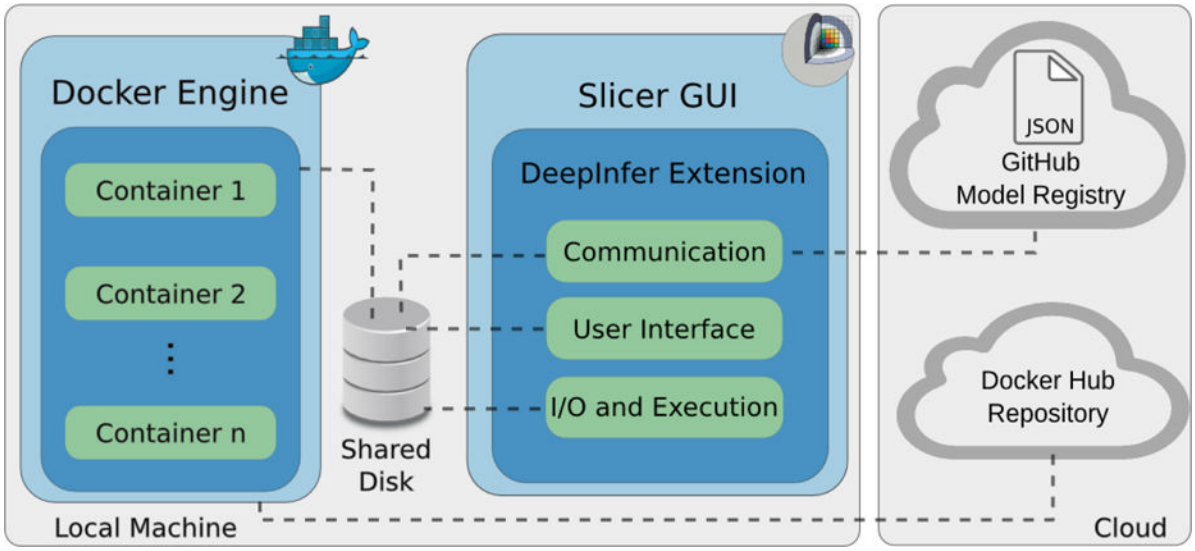

Figure 1.

System architecture: The task-specific models are stored as images on the Docker Hub repository, with their meta-data stored as JSON files on the GitHub model registry. The Slicer module communicates with the model registry to display the available models to the user. The user can select a model for download and deployment. The Docker engine can download the image containing the trained model and run it as a container on the local machine. Once chosen, the Slicer GUI is updated according to the specific model specifications within the JSON model registry file. Additionally, the Docker run command for the container is created. The container reads the input data, processes the data, and writes back the results on the shared disk. Finally, the Slicer display is updated with the inference results.

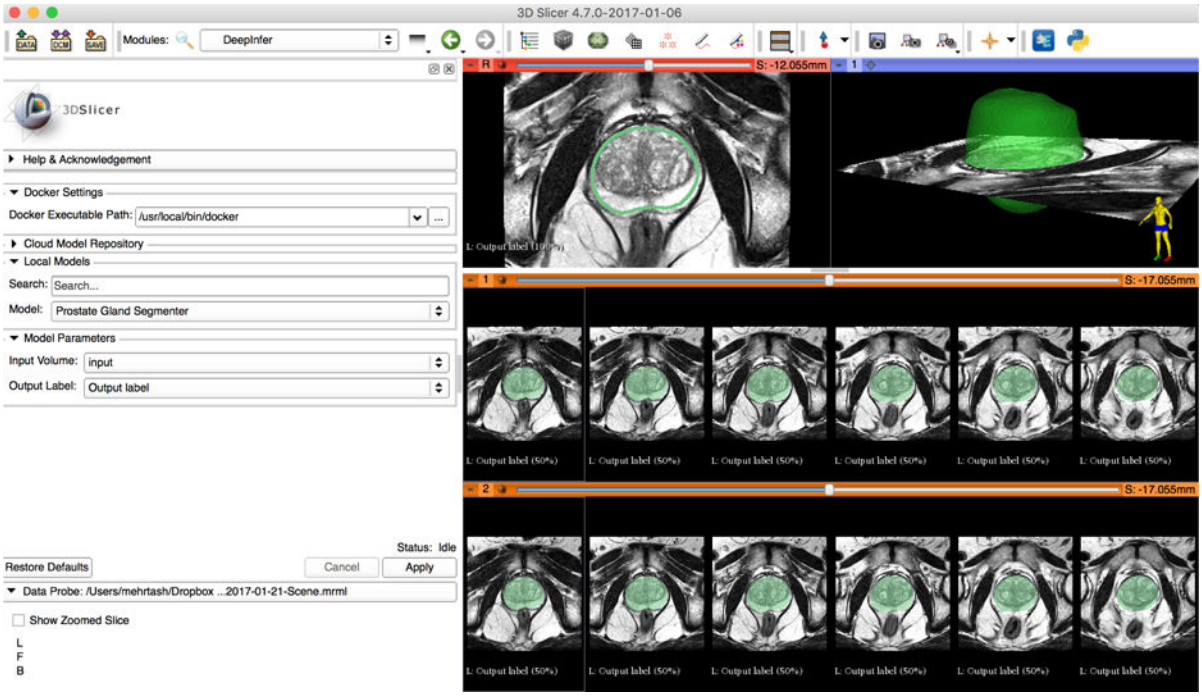

Figure 2.

Visualization of the segmentation results and the GUI of the client side inside 3D Slicer. The left side panel is the GUI which has connection, Model and input/output (IO) sections. The right side panel contains the visualization viewers showing the prostate MRI volume with the delineated prostate gland. The delineation was calculated by a deployed model using our system.

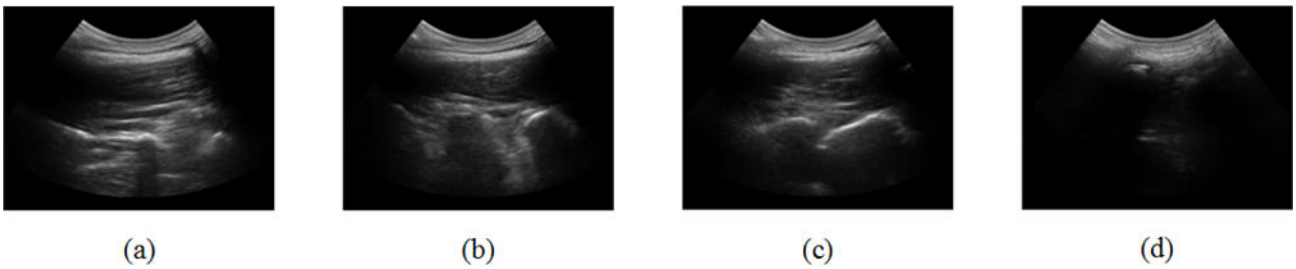

Figure 3.

Signature of (a) transverse processes (b) facet joints (c) laminae and (d) spinous processes in sagittal ultrasound images. Target planes for facet joint and epidural injections are (b) and (c), respectively. Accurate identification of the target plane is challenging, because the spinal structures have similar wave-like structures.

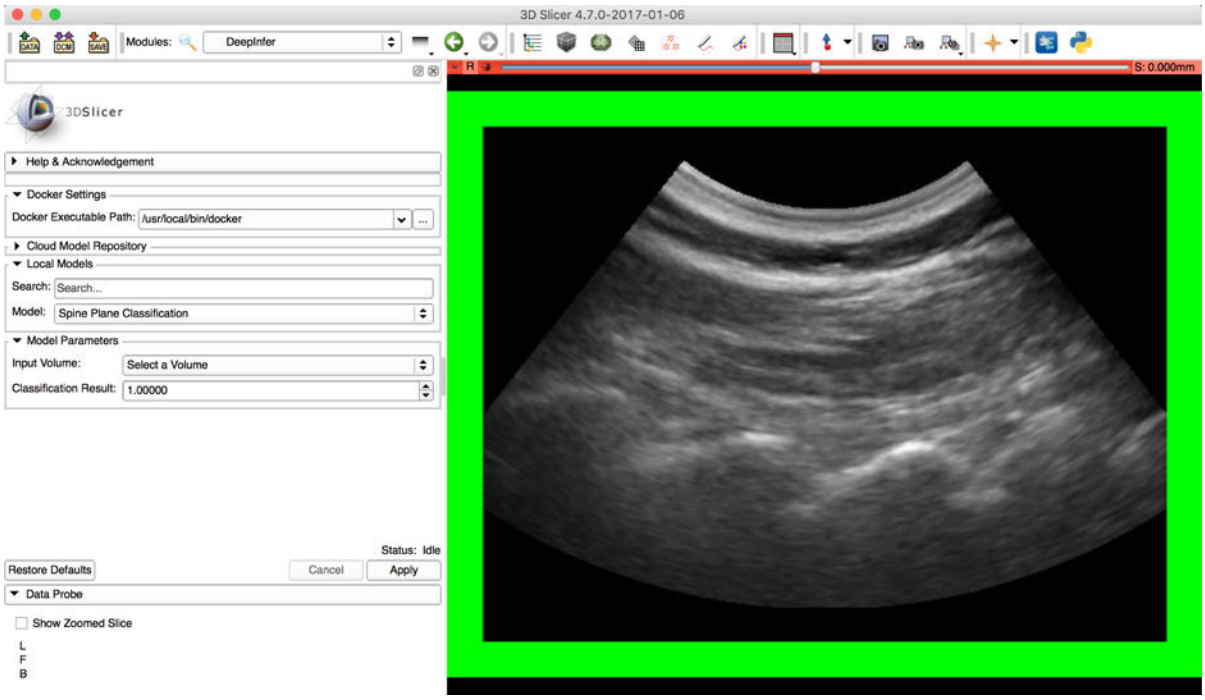

Figure 4.

Visualization of the target plane classification for epidural injection and the user GUI of 3D Slicer. The green frame on the plane and the classification result on the left panel indicate that the current image is a target plane for epidural needle placement.