

Web-Based Spine Segmentation Using Deep Learning in Computed Tomography Images

Young Jae Kim^{1,3,4,*}, Bilegt Ganbold^{2,*}, Kwang Gi Kim^{1,2,3,4}

¹Department of Biomedical Engineering, College of Health Science, Gachon University, Incheon, Korea

²Department of Health Sciences and Technology, Gachon Advanced Institute for Health Sciences and Technology (GAIHST), Gachon University, Incheon, Korea

³Department of Biomedical Engineering, College of Medicine, Gachon University, Incheon, Korea

⁴Medical Device R&D Center, Biomedical & Convergence Institute, Gachon University Gil Hospital, Incheon, Korea

Objectives: Back pain, especially lower back pain, is experienced in 60% to 80% of adults at some points during their lives. Various studies have found that lower back pain is a very common problem among adolescents, and the highest incidence rates are for adults in their 30s. There has been a remarkable increase in using computer-aided diagnosis to assist doctors in the interpretation of medical images. Spine segmentation in computed tomography (CT) scans using algorithmic methods allows improved diagnosis of back pain. **Methods:** In this study, we developed a web-based automatic spine segmentation method using deep learning and obtained the dice coefficient by comparison with the predicted image. Our method is based on convolutional neural networks for segmentation. More specifically, we train a hierarchical data format file using U-Net architecture and then insert the test data label to perform segmentation. Thus, we obtained more specific and detailed results. A total of 344 CT images were used in the experiment. Of these, 330 were used for learning, and the remaining 14 for testing. **Results:** Our method achieved an average dice coefficient of 90.4%, a precision of 96.81%, and an F1-score of 91.64%. **Conclusions:** The proposed web-based deep learning approach can be very practical and accurate for spine segmentation as a diagnostic method.

Keywords: Deep Learning, Computer-Aided Diagnosis, Health Information Systems, Classification, Spine

Submitted: July 24, 2019

Revised: 1st, October 24, 2019; 2nd, January 16, 2020

Accepted: January 18, 2020

Corresponding Author

Kwang Gi Kim

Department of Biomedical Engineering, College of Medicine, Gachon University, 21, Namdong-daero 774beon-gil, Namdong-gu, Incheon 21565, Korea. Tel: +82-32-820-4036, E-mail: kimkg@gachon.ac.kr (<https://orcid.org/0000-0001-9714-6038>)

*These two authors contributed equally to this work.

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

© 2020 The Korean Society of Medical Informatics

1. Introduction

Back pain is one of the most common reasons that patients suffer and a leading cause of disability worldwide [1]. This can be divided into neck pain (cervical), middle back pain (thoracic), and lower back pain (lumbar) based on the region affected. Lower back pain is the most common type of back pain because the lumbar spine supports most of the weight of the upper body. Back pain can originate from the muscles, nerves, bones, joints or other structures of the spine [2]. Diagnostic tests can vary according to their etiology, but usually computed tomography (CT) is considered the gold standard. Nowadays, new technologies are being developing to improve the accuracy of diagnosis and decrease the workload of doctors [3].

Computer-aided diagnosis (CAD) is rapidly entering the

field of radiology and has become a part of routine clinical work for medical image interpretation. The algorithms used in this area consist of several steps that may include image processing, image analysis, and classification with the use of tools, such as deep learning [4]. Deep learning allows computational models composed of multiple processing layers to learn representations of data. Vania et al. [5] used a hybrid method of convolutional neural networks (CNN) for spine segmentation CT scans and achieved good results by comparing this to other methods of spine segmentation. In our study, we developed a web-based automatic spine segmentation system from CT scans using deep learning (CNN), which has been widely used in recent years.

The segmentation process subdivides an image into its constituent parts or objects, depending on the problem to be solved. Segmentation is stopped when the region of interest in a specific application has been isolated [6]. Today, medical imaging modalities generate high resolutions and a large number of images that cannot be examined manually [7]. This is driving the development of more efficient and robust problem-tailored image analysis methods for medical imaging. Automated image segmentation could increase precision by eliminating the subjectivity of the clinician. Most of these methods consist of two steps: identification of the spine and separation of individual segments of the spine vertebrae [8].

There are several advantages to interfacing deep learning to a webpage. We aimed to develop a simple and user-friendly graphical user interface (GUI). Interfacing to a webpage makes it possible to run the method on any operating system (OS) of computers and it provides the opportunity to use any computer using an IP address if it is located on a server. Moreover, the development of the web is improving rapidly, so it will be convenient to use in the future. Deep learning can be classified as supervised and unsupervised. This system will work with the following two types of action. (1) Deep learning supervised method: Supervised learning is conducted to get a desired result (to classify) from any data. Previously prepared test data (prepared data as input and output) should be named as test data and then used to train features by computer. In comparison, unsupervised learning is conducted to get common features from previously unprepared and unclarified data. In our study, we used the supervised learning method. (2) Training specific features of data by computer takes a long time, so it is impossible to train from the very beginning. Therefore, a previously prepared hierarchical data format (HDF5) file is loaded, which makes it easier to re-train [9,10]. When training of this file is finished, it is also saved as a new one. Therefore, it is pos-

sible to use old versions. Previously trained HDF5 images are inserted and then re-training is conducted using requested data.

In this research, we aimed to develop web-based spine segmentation with deep learning using CT scans. The provided model is basically a convolutional auto-encoder, but it has skip connections from encoder layers to decoder layers that are on the same 'level'. There is a general consensus that successful training of deep networks requires many thousands of annotated training samples. Moreover, the network must be fast. Segmentation of a 512×512 image takes less than a second on a recent GPU. We calculated sensitivity, dice similarity coefficient (DSC), precision, recall, and F1-score. In addition, a Bland-Altman plot was drawn with mean difference values for statistical analysis.

II. Methods

1. Dataset

We used 344 images obtained from CT scans of 100 patients for the experiment. Of these, 330 images were used for training, and the remaining 14 were used for testing. Of the 330 images used for learning, 20 were used for validation in the learning process. Three to four CT images were extracted from random locations per patient, and the images used in the test did not overlap with those of patients whose images were used for learning. The CT image size was 512×512 pixels with a value of 12 bits per pixel. We converted the 12-bit value to the 8-bit value and normalized it back to a value between 0 and 1. In this paper, an expert manually annotated the spinal region using the ImageJ digital analyzing tool to obtain the ground truth data. The region of interest obtained through annotation was made with a mask image filled with 0 and 255 in a blank image of the same size as the CT image. For the original image and the mask image, we trained the deep learning model with the original CT image as the input image and the mask image as the output image.

2. Model and Deep Learning

The proposed work was implemented in Python using Keras taking TensorFlow as the backend and trained on a desktop with a 4.20 GHz Intel i7-7700k CPU and a GTX 1080i graphics card with a 32-GB GPU memory; and HTML, CSS, JavaScript, and Bootstrap framework were chosen for the GUI. The Python-based Flask web framework is being developed progressively, and there were only a few problems during coding. Also, Python 3.6, TensorFlow 1.8.0, Flask 1.0.2, and Keras 2.2.0 were used for development.

The function of the web system and training of spine segmentation with deep learning have the following steps. To develop the web system, we decided on the Python programming language, which was also the most suitable language for training. We used the U-Net architecture to train the data. U-Net is a convolutional network architecture for the fast and precise segmentation of images, and it was helpful for our study because it is an open-source architecture [11,12]. Up to now, it has outperformed the prior best method (a sliding-window convolutional network) on the ISBI Challenge Workshop for the Segmentation of Neuronal Structures in Electron Microscopic Stacks. Also, there are many frameworks, such as Theano and CNTK for the backend in Keras but we chose TensorFlow, which is the most successful framework that is being developed.

In the illustration of the U-Net architecture shown in Figure 1, each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote various operations:

$$Epochs = \frac{Input\ images}{Batch\ size}, \text{ max batch size} = 512.$$

If the number of input images is 1,000, then the batch size is 32. It is perfectly configured.

3. Web Framework

Flask is a micro web framework written in Python [13,14]. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies, and several common framework-related tools. Extensions are updated far more regularly than the core Flask program. There are many advantages of a web-based system in comparison to a desktop application. For example, if a web system is deployed on any server, a user can see the results with the help of a computer from anywhere, regardless of the operating system of the computer. Moreover, there is no need for installation, and there is no special requirement for computer specifications. Besides, there are a lot of web frameworks, but Flask does not require precise capability so we would not face any problems in developing the proposed approach.

4. Web-Based Spine Segmentation Using Deep Learning in CT Scan Images Architecture

We uploaded similar CT data with a label that would lead to certain results with the supervised method of deep learning or training according to the prepared data. In this case, we

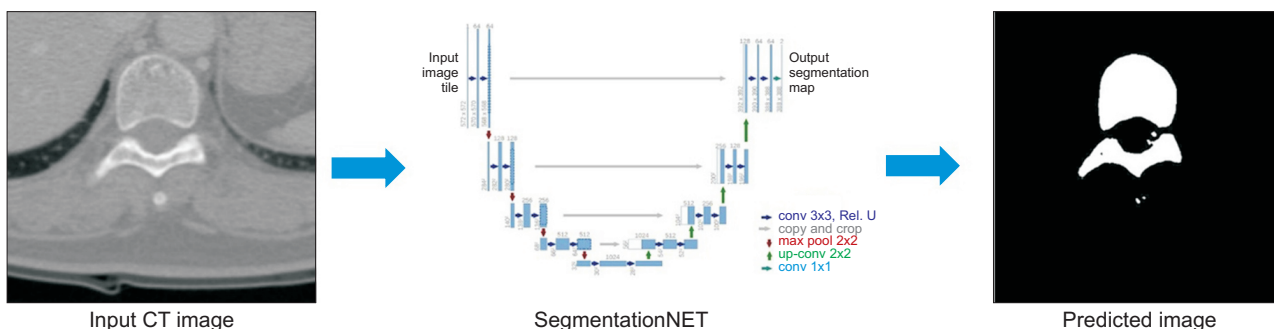


Figure 1. Process of spine segmentation using the U-Net architecture. CT: computed tomography.

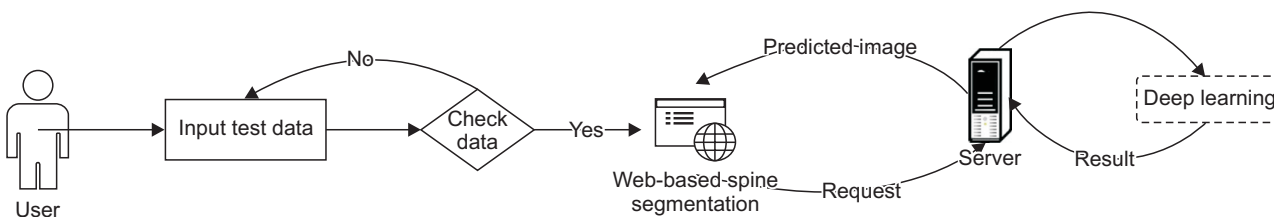


Figure 2. Web-based spine segmentation process.

can estimate the dice coefficient from those image results. If a user wants to train using the supervised method, both the CT data and label (mask image) must be uploaded. Therefore, a prepared HDF5 file is loaded, and then we can see the result of the test CT data. Additionally, we can calculate the dice coefficient by comparing the uploaded and predicted images, which shows the similarities between the images.

First, data that a user wants to predict should be uploaded to the web-page. Then, if there is no problem in uploading the data, a request will be sent to the server. A NumPy file of those images will be generated, and if there is no error, data will be predicted using the deep learning model, and it will be shown as a JPEG file [15] (Figure 2).

III. Results

The automatic segmentation method was evaluated in terms of sensitivity, DSC, precision, recall, and F1-score using the data gathered [16,17]. For evaluation, we obtained true positive, true negative, false positive, and false negative results using the pixels of segmented images by automatic segmentation, and these results are shown in Table 1. The results of automatic segmentation by deep learning are shown in

Table 1. Comparison of sensitivity, dice similarity coefficient (DSC), precision, recall, and F1-score between the results of manual and the deep learning measurements

Sensitivity	DSC	Precision	Recall	F1-score
86.99	90.40	96.81	86.99	91.64

Figure 3. In the Bland–Altman plot, the mean difference was 1783.36, as shown in Figure 4.

Figures 5 and 6 show the user interface (UI) of a web-based system. Figure 5 shows the UI for uploading a file on a web-page. Figure 6 shows the result of segmenting the spine region through deep learning by sending the uploaded image file to the server.

IV. Discussion

In this paper, we proposed a web-based CT image segmentation system that uses deep learning. The proposed approach sends a CT image received through the web to the server, segments the spinal region through deep learning on the server, and provides the result back via the web. In this way,

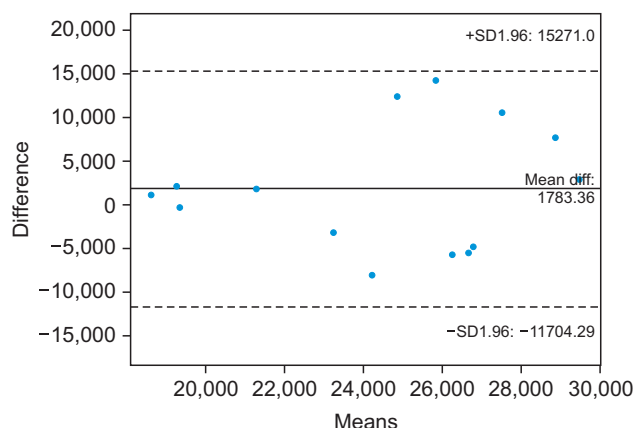


Figure 4. Bland–Altman plot between deep learning-based segmentation and manual segmentation results.

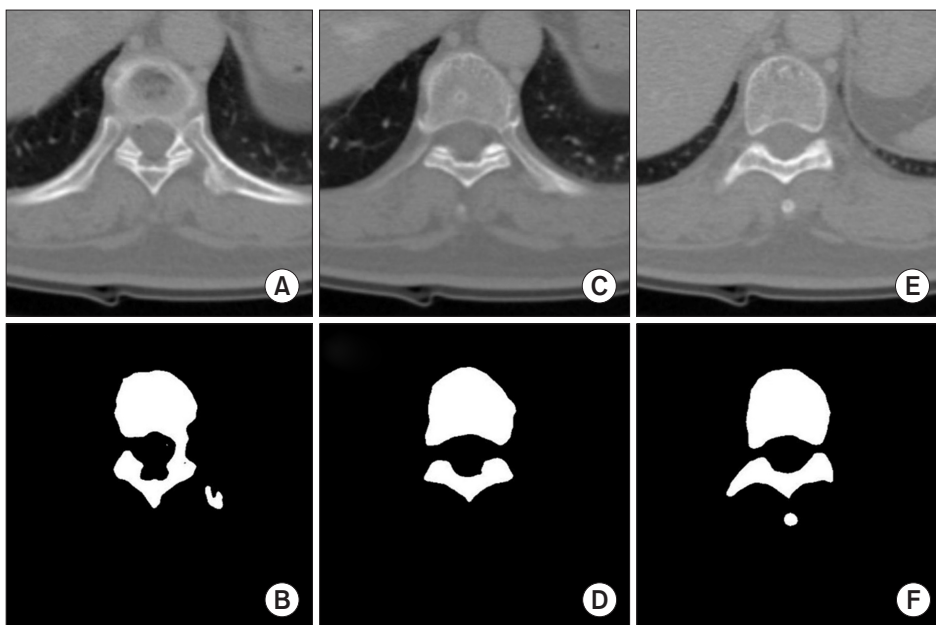


Figure 3. Examples of spinal area segmentation results based on deep learning: (A, C, E) computed tomography images and (B, D, F) deep learning-based spinal segmentation results.

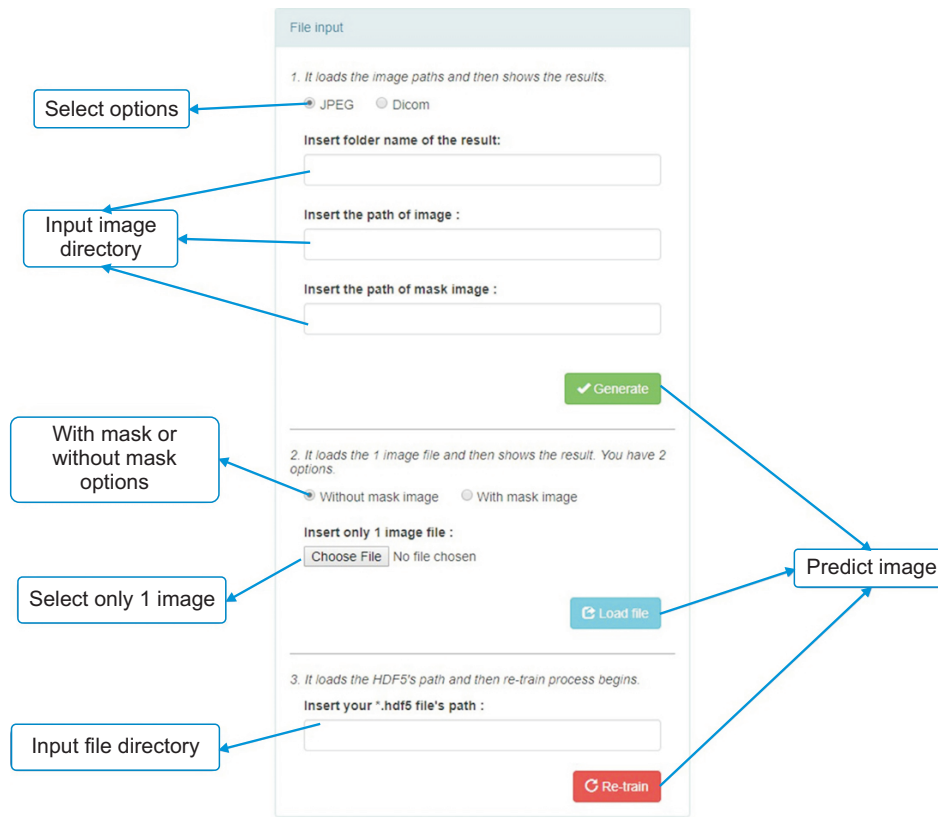


Figure 5. User interface of webpage for uploading files.

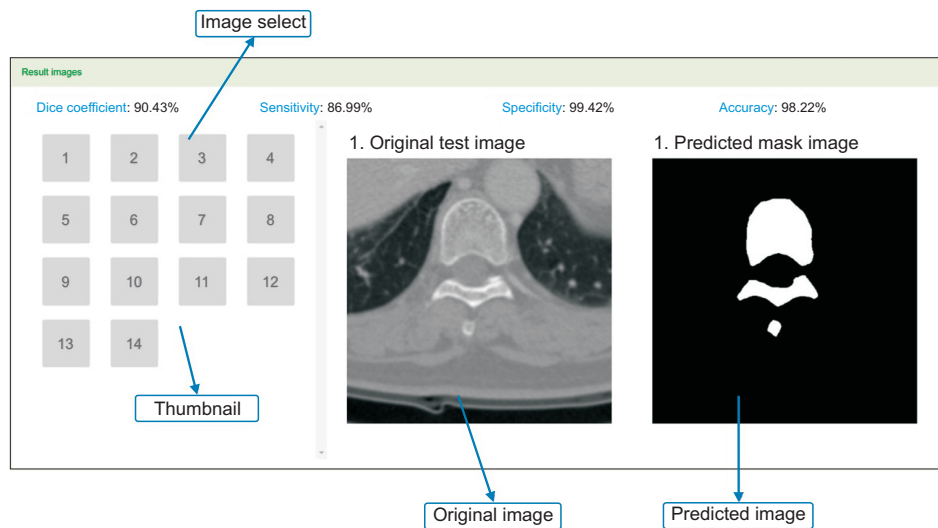


Figure 6. User interface of webpage to provide spinal segmentation results using deep learning.

CT image segmentation results obtained using the GPU-based deep learning model could be provided easily and quickly anywhere on the web regardless of the specifications of the local computer. Our proposed web-based system has strong advantages in terms of scalability and accessibility. In other words, the process of entering and learning data is the same, so it is easy to expand into web-based systems for various organs or lesions. Additionally, because it is provided over the network, anyone who is allowed on the network can

use the system anywhere, regardless of computer specifications, through a web browser.

However, the system requires the resolution of some problems. Although this system excluded security issues because it was developed for research purposes, the addition of security transmission modules is essential for future system deployments because medical data is delivered over the network. In addition, various additional studies and verification of deep learning-based spinal segmentation results are

required. To develop a more accurate deep learning model, it is necessary to collect additional data and to select optimal parameters through comparison with various deep learning models and parameters. It is also necessary to verify the clinical significance and reliability of the model by performing multi-instrument validation through images obtained from various medical institutions and equipment. If the security issues for medical data transmission are addressed first, the web-based system proposed in this paper can be effectively used for multi-center verification.

If some problems are solved through further research and verification of the system proposed in this paper, it is expected that the various information needed by clinicians can be provided through the web through accurate spinal area segmentation in the future. It is also expected that various deep learning-based algorithms will be developed and provided via the web in accordance with clinical needs, and it is expected that they can be conveniently and easily applied for clinical use without additional purchase of expensive equipment.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgments

This research was supported by the GRRC program of Gyeonggi Province (No. GRRC Gachon 2017-B01), the Gachon University Research Fund of 2019 (No. GCU-2019-0369), and the Ministry of Science and ICT, Korea, under the Information Technology Research Center support program (No. IITP-2019-2017-0-01630) supervised by the Institute for Information & communications Technology Promotion (IITP).

ORCID

Young Jae Kim (<http://orcid.org/0000-0003-0443-0051>)

Bilegt Ganbold (<http://orcid.org/0000-0002-7136-1271>)

Kwang Gi Kim (<http://orcid.org/0000-0001-9714-6038>)

References

1. Mayo Clinic. Back pain [Internet]. Scottsdale (AZ): Mayo Clinic [cited at 2020 Jan 31]. Available from: <https://www.mayoclinic.org/diseases-conditions/back-pain/symptoms-causes/syc-20369906>.
2. Abu-Naser SS, Aldahdooh R. Lower back pain expert system diagnosis and treatment. *J Multidiscip Eng Sci Stud* 2016;2(4):441-6.
3. National Academies of Sciences, Engineering, and Medicine. Improving diagnosis in health care. Washington (DC): National Academies Press; 2015.
4. Shiraiishi J, Li Q, Appelbaum D, Doi K. Computer-aided diagnosis and artificial intelligence in clinical imaging. *Semin Nucl Med* 2011;41(6):449-62.
5. Vania M, Mureja D, Lee D. Automatic spine segmentation using convolutional neural network via redundant generation of class labels for 3D spine modeling [Internet]. Ithaca (NY): arXiv.org; 2017 [cited at 2020 Jan 31]. Available from: <https://arxiv.org/abs/1712.01640>.
6. Kamboj A, Gupta A. Simulink model based image segmentation. *Int J Adv Res Comput Sci Softw Eng* 2012;2(6):146-9.
7. Lathen G. Segmentation methods for medical image analysis: blood vessels, multi-scale filtering and level set methods [master's thesis]. Linkoping, Sweden: Linkoping University; 2010.
8. Nyul LG, Kanyo J, Mate E, Makay G, Balogh E, Fidrich M, et al. Method for automatically segmenting the spinal cord and canal from 3D CT images. In: Gagalowicz A, Philips W, editors. *Computer analysis of images and patterns*. Heidelberg, Germany: Springer; 2005. p. 456-63.
9. Folk M, Heber G, Koziol Q, Pourmal E, Robinson D. An overview of the HDF5 technology suite and its applications. *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*; 2011 Mar 25; Uppsala, Sweden. p. 36-47.
10. Collette A. *Python and HDF5: unlocking scientific data*. Sebastopol (CA): O'Reilly Media Inc.; 2013.
11. Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells W, Frangi A, editors. *Medical image computing and computer-assisted intervention – MICCAI 2015*. Cham: Springer; 2015. p. 234-41.
12. Dong H, Yang G, Liu F, Mo Y, Guo Y. Automatic brain tumor detection and segmentation using U-net based fully convolutional networks. In: Valdes Hernandez M, Gonzalez-Castro V, editors. *Medical image understanding and analysis*. Cham: Springer; 2017. p. 506-17.
13. Grinberg M. *Flask web development: developing web applications with python*. Sebastopol (CA): O'Reilly Media Inc.; 2018.

14. Mainland G, Morrisett G, Welsh M, Newton R. Sensor network programming with Flask. Proceedings of the 5th International Conference on Embedded Networked Sensor Systems; 2017 Nov 6-9; Sydney, Australia. p. 385-6.
15. Van der Walt S, Colbert SC, Varoquaux G. The NumPy array: a structure for efficient numerical computation. *Comput Sci Eng* 2011;13(2):22-30.
16. Zou KH, Warfield SK, Bharatha A, Tempany CM, Kaus MR, Haker SJ, et al. Statistical validation of image segmentation quality based on a spatial overlap index: scientific reports. *Acad Radiol* 2004;11(2):178-89.
17. Davis J, Goadrich M. The relationship between precision-recall and ROC curves. Proceedings of the 23rd International Conference on Machine Learning; 2006 Sep 10-13; Sofia, Bulgaria. p. 233-40.