2018

Web-Based, Deep Learning Assisted Medical Image Tagging Tool

Nicolás Arias González
Grand Valley State University

Web based, Machine Learning Assisted Tool for Medical Image Tagging

By
Nicolás Arias González
December, 2018

Web based, Machine Learning Assisted Tool for Medical Image Tagging

By

Nicolás Arias González

A project submitted in partial fulfillment of the requirements for the degree of

at

Grand Valley State University

December, 2018

Jonathan Engelsma **12/11/2018**

Table of Contents

Abstract

One of the biggest challenges when building supervised machine learning models is to obtain the desired dataset along with its respective annotations. This is especially true in the medical field where all data produced is expected to be consumed by a human being instead of a machine. More often than not, the data can be found only by itself and data scientists are burdened with the task of manually creating the tags for it, a tedious and time-consuming task.

This project aims to speed up the process of manually annotating regions of interest (ROI) in images from computed tomography (CT) scans by leveraging fully convolutional deep networks and web technologies. A partially trained deep learning model suggests ROI to the user who evaluate and adjust them. These corrected images can then be fed to the model as ground truths to continue training. The end result of this process is the tagged dataset and a fully trained machine learning model for predicting ROI in CT scans. In an experiment performed with the help of a medically trained volunteer, tagging images aided by a model trained with 3% of the dataset resulted in a 7x speedup over the manual process.

Introduction

The algorithmic nature of the medical field makes it very data centric. Every decision taken by a medical professional should be backed by a piece of information about their patient, whether it is qualitative, such as their skin tone or general appearance or quantitative, like their heart rate, respiratory rate or the result of a blood screen, data is ubiquitous in this domain. Due to this fact, computer scientists have taken an interest in improving medical processes with the aid of computers since the early days of artificial intelligence, first by attempting to translate human thought processes to algorithms in the form of expert systems and more recently by finding underlying correlations through the use of neural networks. As machine learning techniques have become more sophisticated, the results from applying them to medicine have reached impressive levels of accuracy in studies with a wide range of applications like insurance fraud prediction, tumor identification in images or genetic analysis just to name some examples.

Machine learning techniques are generally divided into two categories: unsupervised and supervised learning. The difference between them being that the latter, which will be the focus of this project, requires each piece of data to be accompanied by “ground truth” annotations, representing what the model is expected to output when presented with each data point. This program will present the users with an easy to use web interface to annotate structures in CT scans, a deep learning model will assist the process by generating predicted annotations which the user will then correct or accept without changes, reducing the time necessary to tag large volumes of data.

A simple experiment with one participant was performed to assess the effectiveness of the tool for annotating the lungs in thoracic CT scans. One of the main reasons for the lungs being chosen as the structure for evaluating the proposed system is that even though it doesn’t require much attention in terms of identifying the structure, it is a very time-consuming task because they are large in comparison with other elements and appear in most of the slices in a thoracic scan, it is expected that annotating smaller structures will take less time.

The tool is intended to be used in the future to train a machine learning model that identifies lesions that limit the usable volume within the patient’s lungs such as ground glass opacity, emphysema or pulmonary fibrosis in thoracic CT scans, however, it is generic enough that it can be used to annotate any kind of structure in different types of CT scans, provided that they are supplied to the program in DICOM format, a standard format for storing medical images.

Problem Statement

Tagging datasets of CT scans usually requires manual review and annotation of each of the slices that compose the scan by at least one radiologist, an expensive and time-consuming task. This project aims to speed up the process of annotating ROI in CT scan images by providing medical professionals with an easy to use web application assisted by a partially trained fully convolutional neural network (FCN).

Background and Related Work

Manual image segmentation tasks are typically done with at least one of three different approaches: A polygonal approach, where the regions of interest are delineated by using either bounding boxes or complex polygons, a pixelwise approach typically achieved with a brush like functionality in which the user "paints" over the region of interest and a superpixel approach that attempts to speed up the process by allowing the user to tag clusters of similar pixels at the time. All of these approaches can be complemented by using either algorithmic or machine learning based predictions to provide the user with some base prediction of the labels.

Since the most basic form of semantic segmentation consists of just recording two points for two corners of a bounding box, some teams opt for building their own solution that produces the data in the format they need it instead of looking for a ready-made solution. A number of the open source tools available for the task are the product of the work of those teams. This is the case of tools such as the COCO-stuff annotator (1) written in Matlab for the COCO-stuff dataset for everyday object classification or the commacoloring (2) project that uses a simple web interface to generate dense image masks for training autonomous vehicles. In addition to these ad-hoc solutions, there are some software as a service (SaaS) offerings and for medical images, there are some sites that offer professional tagging services (3). Of the SaaS solutions, two stand out for the task: Labelbox (4), a general purpose image and video tagging tool provides the user with the possibility to generate text, polygon and pixel annotations for their data, and Supervisely (5), which provides not only data tagging options but also has functionalities for data transformation and machine learning model training, going well beyond the scope of this project. None of these tools are specialized in annotating medical images, although it is possible to use them with this purpose by converting the images, typically distributed in DICOM format, to JPEG or PNG files. It isn't common for software specialized in processing medical images to be web based. Of the examples evaluated, the ones that stood out the most were Harvard University's 3d Slicer (6) and ITK-Snap (7). Both are exceptional pieces of software mostly focused on algorithmic segmentation but are fairly complex and difficult to use.

Program Requirements

A user will be able to quickly generate masks for segmenting the lungs in CT scans through the web application with the aid of the deep learning model. She will be presented with a single page application that consists of three sections: A list of the available scans, a work area and a toolbar. Using the options available in the toolbar, the user should be able to draw or erase the mask, clear the canvas for the selected slice, manipulate the brush size or the zoom and copy the mask from the previous slice. This last functionality will be useful when performing unassisted tagging as the difference in the regions of interest between slices is usually small.

The data annotated by the user using this tool will be used as ground truth annotations for training the neural network, resulting in a positive feedback loop where the user's productivity is increased due to the network becoming more accurate in generating predictions. When the user finishes tagging all of the data, they will also have a fully trained model ready to be used in practical applications.

A typical tagging workflow in the application starts with the user selecting a scan from the list, this action loads the first slice of the scan into the canvas, enables the slice navigation strip and the toolbar. If a mask has already been created either manually or by the neural network, the user can choose to correct it or clear the screen to start manual tagging. When a user changes the mask in any way and selects a different slice to work with, the changes are automatically sent to the server, the process is repeated until all of the images in the scan have been tagged.

Implementation

A total of 2655 Thoracic and Thoracoabdominal CT scans in DICOM format were obtained from three different sources: 1013 from the Lung Image Database Consortium image collection (8) (5 out of the 1018 total failed to download), 50 from Cornell University's VIA-ELCAP public lung database (9) and the remaining 1592 were obtained from Kaggle's Data Science Bowl 2017 competition (10) which unfortunately are not available at the time of writing because of dataset usage restrictions, as they state in their website.

The images were extracted from each DICOM file: An outline of the body that was later used in post processing to remove false positives, an algorithmic segmentation of the lungs that was used as a starting point for the annotations and a PNG file with the contents of the original image for training the model.

Deep Learning Model

The deep learning model used in this project is inspired by University of Freiburg computer vision group's U-NET architecture (11), which has been trained successfully to identify structures within medical images and works remarkably well even with few data points. This U-NET is an example of a fully convolutional neural network or an encoder-decoder network, typically used in semantic segmentation of images.

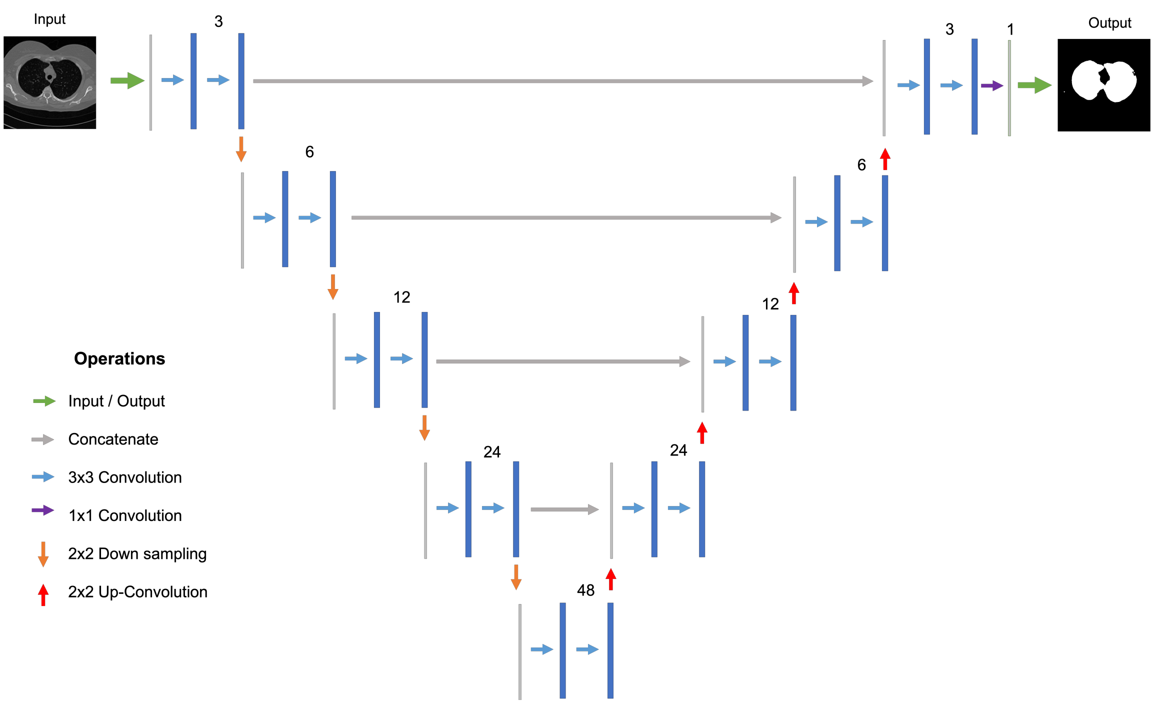

Figure 1 - Deep learning model architecture

Architecture

Even though the implemented network is inspired by U-NET, its architecture is different from the one proposed in the original paper in two main aspects, the number of filters in each convolutional layer is greatly reduced and the image size is kept constant by applying padding to each convolutional layer.

The contracting path consists of four sequences of two 3x3 convolutions and followed by a 2x2 max-pool operation. The number of filters in the convolutional layers starts at 3 and it is doubled after the max-pool up to a maximum of 48. After reaching this point, the expanding path starts with an "up-convolution" operation, which is performed via a 2x2 up sampling operation followed by a 3x3 convolution. The result from the opposing part of the contracting path is concatenated to the result of the up-convolution and two 3x3 convolutions are applied to it. The number of filters in the convolutional networks of the expanding paths are halved after each up-convolution to a minimum of 3. The result from this last layer is passed to the output layer, that also a convolutional layer with the number of filters equal to the number of classes to be predicted, a 1x1 kernel size and a sigmoid activation function. All of the convolutional layers except for the output layer have a Rectified Linear Unit function (ReLU) as their activation function, they are padded in order to keep the image size constant across the network and the kernel weight matrix is initialized using a random normal distribution.

Values obtained from the network are matrices with WxHxc dimensions, where W is the width of the input image, H is its height and c is the number of classes the network predicts. The values Pijc contained within said matrices represent the probability of a pixel with x,y coordinates equal to i,j of being of class c. Since the implemented network currently predicts only one class from grayscale images, both the input and output values have 512x512x1 dimensions.

In order to avoid overfitting, there was an attempt to add dropout operations throughout the network but this proved ineffective because of a disbalance between the images that don't contain lungs, i.e. their mask is completely black, and those that did, resulting in a network that only predicted black images but reported an accuracy of about 85%. This idea will be revisited in the future by obtaining a random sample of images with blank masks equal to the number of images with non-blank masks.

Training

An algorithmic solution for lung segmentation was developed both for comparison purposes and to speed up the initial generation of ground truth masks for the purpose of training the network. The solution consisted of applying a watershed transformation to the darkest regions within the body in order to identify their borders, this process was extremely effective at identifying the contours of the lungs and was described by the volunteer in the experimental test as being almost perfect, but the main drawback it had is that it was sensible to all the areas in the body that contain air, which are not limited to the lungs but also include the trachea, stomach and colon. For this reason, every image had to be reviewed and these structures removed from the masks. A total of 14803 images from 62 scans were annotated using this method and they were split into training, validation and testing sets at a ratio of 60:20:20 respectively, no dataset augmentation was applied. The model was then trained using an Adam optimizer with a learning rate of 0.0001 and using binary cross entropy as the loss function, this loss function was monitored for the validation set for improvements between epochs and if it failed to improve for three consecutive epochs the training process stopped early.

Set	**Accuracy**
Training	99.61%
Validation	99.60%
Testing (With post-processing)	97.53%

Table 1 - Training results of the best model obtained

It is expected that with more annotated data, the generalization capabilities of the network improve and the accuracy of the predictions on previously unseen data gets better as well. For the purpose of this project, the results are good enough to generate masks that can be corrected by a user and fed back into the train-assistance loop.

Post-processing

From visual inspection of the neural network predictions presented two issues that were easily mitigated in post processing. The first one was false positive predictions outside of the patient's body. This was fixed by algorithmically obtaining a binary mask of the contour body and performing an elementwise multiplication between the mask and the prediction. The mask generation process for the body contour takes advantage of the fact that the body is surrounded mostly by air in the scan and that it is the largest enclosed area in the image. Slices can present issues in slices closer to the head can present issues because the body can appear to be segmented into up to three different parts: both shoulders and the head but these images don't contain relevant data for lung analysis so they can be safely ignored in this instance.

The other issue that the predictions presented was that the mask would follow the contour of the lungs but they would often stop before reaching the limit for the lung parenchyma, a binary dilation transformation was applied to the predicted mask in order to expand the area outward and make sure that the lung area was covered. The best result in the experimental evaluation was obtained with this method as it required fewer corrections from the user.

Front End

Written in React (12) and relying heavily in the use of HTML5 canvas, the front end provides the user with a minimal set of tools for them to generate the image masks for the ROI, it consists of a paginated list of scans on the left hand side, a toolbar on the right hand side, the canvas at the center and underneath it a navigation strip. Keyboard shortcuts are implemented to encourage the user to use both hands and keep the mouse cursor inside the canvas area for as long as possible.

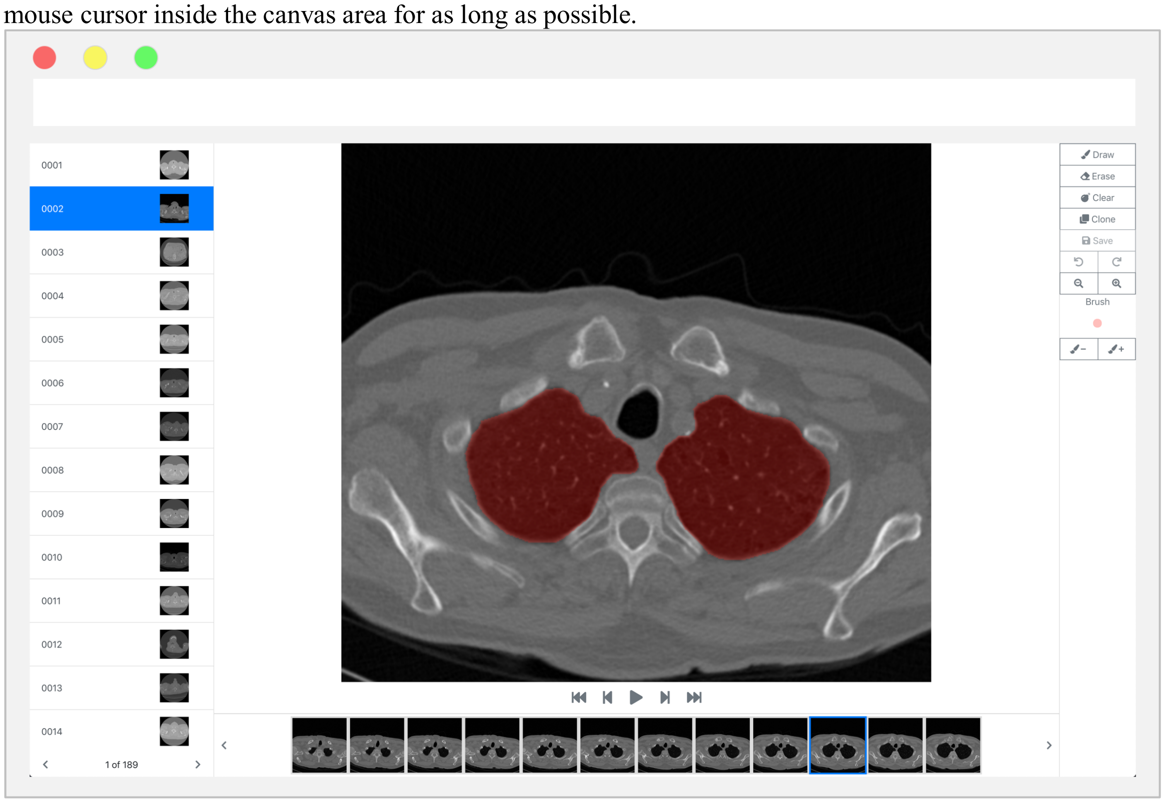

Figure 2 - Front End Graphic User Interface

Back End

A RESTful back end written in python with the flask microframework (13) is responsible for managing all of the scan and slice related operations, providing the list of scans and slices per scan, reading the DICOM files and transforming them into PNG files that can be interpreted by the browser and processing the mask images produced by the user and associating them with its corresponding slice.

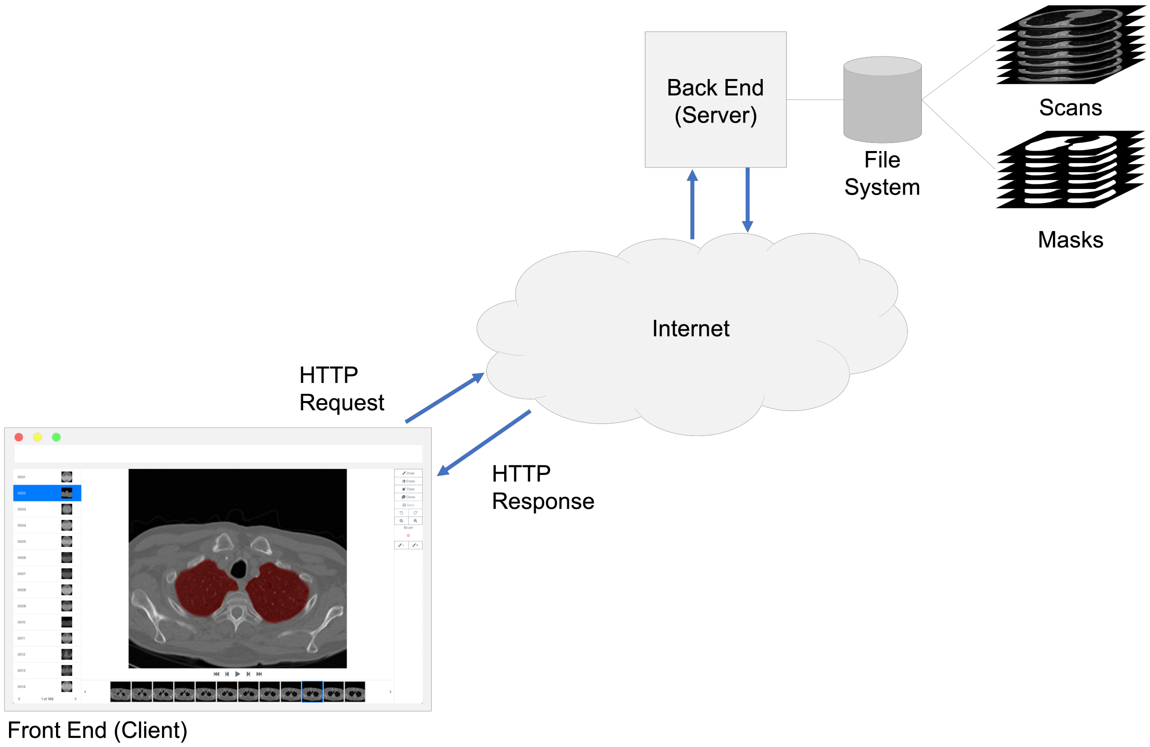

Figure 3 - Web Architecture

Results, Evaluation, and Reflection

A medical doctor trained in reading CT scans volunteered to test the effectiveness of the tool in speeding up the process of tagging the lungs in CT scans. She was given instructions on how to use the web interface and all of the implemented functionalities, then she was allowed to familiarize herself with the controls, keyboard shortcuts and tag images without any measurements being performed in order to avoid having apparent improvements between tests because of increased proficiency with the use of the tool. After the volunteer had familiarized herself with the application, she was instructed to tag fifty slices of the same scan with four different types of assistance.

No assistance (Control): The volunteer performed the test using only the application functionalities. As the differences between consecutive slices are relatively small, she relied heavily in the use of the clone functionality, which copies the mask from the previous slice into the current one and expanded or contracted the mask using the brush tool to accurately annotate the lungs.

Edge detection assistance: A prediction of the lung mask was provided by using the watershed transformation to identify the darkest areas within the body. This approach is highly effective and was qualitatively described by the volunteer as being almost perfect in identifying the lungs but it produced false positives that had to be removed in other areas of the body that typically contain air such as the trachea, the stomach and the colon.

Machine learning assistance: The lung masks were predicted by the neural network and the subject had to remove false positives and correct false negatives from the prediction.

Machine learning assistance with post-processing: After predicting the mask using the machine learning model, a binary dilation transformation and any positive predictions outside or at the contour of the body were automatically discarded.

Assistance	Time	Speedup
No Assistance (Control)	51.26	-
Edge Detection	7:10	7.17
Machine learning	20:40	2.49
Machine learning with post-processing	7:09	7.19

Table 2 - Experimental Results

Conclusions and Future Work

Given the amount of data that was used to train the model compared to what was available, the results of the experiment are remarkable, the positive feedback loop between the tagging user and the neural network worked as predicted and the expectation is that the process of tagging the remaining available dataset should continue to improve as more of the data is used for training the network. It could be argued that the algorithmic solution developed serves the same purpose as the neural network to segment the lungs and it doesn't require any prior training, but it consistently identifies false positives and it is not a general solution for the segmentation of other structures as it would need specific code for each of the elements of interest, an extremely difficult or even impossible task in some cases.

In terms of the machine learning model, the next step would be to procure scans of patients that present pulmonary fibrosis, ground glass opacity and emphysema lesions and expand the model to recognize them, transfer learning can be used for this task in order to take advantage of the existing network structure and its already trained weights.

One opportunity to improve the user interface would be to make a mobile application for tablet devices and take advantage of touch inputs with stylus pens to make the drawing process more natural for users, the fact that the web application was written in React makes it a prime candidate for a react-native adaptation. The current backend served its purpose for this experiment, but it really is a prototype of what the real implementation should be. The library code for manipulating the DICOM files and generating images can be preserved but the rest should be rewritten in order to add crucial functionalities such as the addition of a scan upload functionality and training scheduling, versioning and evaluation of the machine learning model.

As an initial evaluation for a much bigger product, this project is considered a complete success. Both the deep learning model and the process as a whole were validated as an effective solution to the problem at hand and a clear path was delineated for future iterations of the development process, this project is expected to become the basis of an entrepreneurial venture in Colombia in the coming year.

Bibliography

1. COCO-Stuff Annotation Tool [Online] H. Caesar, J. Uijlings, V. Ferrari [Cited: December 11, 2018.] https://github.com/nightrome/cocostuff10k#annotation-tool
2. Commacoloring [Online] comma.ai [Cited: December 11, 2018.] https://commacoloring.com
3. Cogito Tech [Online] Cogito Tech LLC [Cited: December 11, 2018.] https://www.cogitotech.com/services/healthcare-training-data/
4. LabelBox [Online] Labelbox Inc [Cited: December: 11, 2018] https://labelbox.com/product
5. Supervisely [Online] Deep Systems LLC [Cited: December 11, 2018] https://supervise.ly/product
6. 3D Slicer [Online] BWH Harvard University [Cited: December 11, 2018] https://www.slicer.org/
7. ITK-Snap [Online] ITK-SNAP [Cited: December 11, 2018] http://www.itksnap.org/pmwiki/pmwiki.php
8. LIDC-IDRI [Online] The cancer imaging archive [Cited: December 11, 2018] https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI
9. VIA-ELCAP [Online] Cornell University Public Lung Image Database [Cited: December 11, 2018] http://www.via.cornell.edu/lungdb.html

10. Data Science Bowl 2017 [Online] Kaggle Inc [Cited: December 11, 2018] https://www.kaggle.com/c/data-science-bowl-2017
11. Ronneberger O., Fischer P., Brox T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham
12. React JS Framework [Online] Facebook Inc [Cited December 11, 2018] https://reactjs.org/
13. Python FLASK Microframework [Online] Armin Ronacher [Cited: December 11, 2018] http://flask.pocoo.org/