Detecting Brain Scan Anomalies using U-Net Based Architectures

UY, Mark Christopher
HKUST
mcsuy@connect.ust.hk

ANG, Clyde Wesley Si
HKUST
cwsang@connect.ust.hk

Abstract

Advancements in neural network models have enabled computers to classify images more accurately than human beings [1] and produce human-like results in other computer vision tasks. In this paper, we explore the application of image segmentation onto the medical field, as we aim to classify each pixel of a brain MRI scan into two categories: one for pixels that contain tumors, and one for pixels that do not contain tumors. We applied the U-Net Architecture [2] and explored variations that involve adding an attention mechanism or a recurrent structure to see whether some models are able to more accurately partition the image. Finally, we discovered that adding an attention mechanism to the U-Net Architecture can achieve better results, while adding recurrent structures seem to result in worse partitions.

1. Introduction

The past few years have seen an explosion of innovation and huge shifts of paradigm in the field of computer vision. Deep learning methods, once unknown and on the fringe, have become the de facto standard in almost all computer vision tasks such as object recognition, object classification, facial recognition, scene detection, and image segmentation [3]. One large and important area of computer vision that stands to gain a lot from the deep learning revolution is medical imaging. In this sub-field of computer vision, researchers not only encounter the problems found in general computer vision, but also have to deal with the problem of low quality images [4]. As a result, deep learning models, which are very powerful learners, are more suited towards medical imaging when compared to traditional machine learning methods.

One important task in this sub-field is image segmentation of brain MRI scans into tumors and non-tumors. This is interesting because cancer detection is a difficult medical task that can potentially benefit a large population. If the proposed variations of the U-Net model [2] can produce reliable results, then it can be used as a basis for further medical research. In particular, we will explore adding a recurrent structure or an attention mechanism to the U-Net model, as these modules have been shown to increase accuracy in many natural language processing tasks [5]. By applying such variations, we are able to test whether these modules are also relevant to the field of medical imaging, which can influence future research ideas.

In this paper, we will first review related literature to medical image segmentation, describe the dataset used in the project, and discuss the key characteristics of the dataset. We will also walk through the pre-processing done on the original images, the specific details of each model variation, and the results of the experiments. The main results of the paper show that attention mechanisms are able to get better results, as the model seems to be able to infer more information from surrounding pixels, while adding a recurrent structure leads to worse results, as we are introducing unnecessary complications to the model.

2. Related Work

In this section, we discuss some related literature in image segmentation, including both conventional methods and neural network models. We then discuss the key difference between our models and previous papers.

2.1. Conventional Methods

We first discuss conventional methods in image segmentation. As our model uses a neural network, there exists a significant difference in model paradigm. However, these methods are still worth mentioning due to their influence in image segmentation.

2.1.1 Thresholding

The simplest method in image segmentation is the thresholding method [6]. The idea is that given a gray-scale image, we want to turn it into a black-and-white image, which represents a partition of the original image. This can be done by a simple threshold on the original gray-scale pixel values, or we can employ more complicated methods that consider surrounding pixels, such as Otsu thresholding [7].

Some papers also propose implementing a threshold on the radiographs from CT scans instead of the reconstructed images [8], but the method used is still conventional thresholding schemes.

Although these models can be tuned to perform well under specific circumstances, the results can be unreliable as there is a lack of contextual understanding between pixels [9]. Additionally, thresholding methods that consider surrounding pixels somehow employ an attention mechanism, as each surrounding pixel is allocated a weight that depends on its distance from the current pixel. However, this is different from the attention mechanism used in our models since our attention mechanism is more robust, allowing for high dependency between pixel values and attention weights.

2.1.2 Clustering

Another commonly used method in image segmentation is clustering. The idea is that pixels in the same category will have similar features, and can be clustered in the same group based on its distance from a set of representative pixels [10]. So if we are given k classes of pixels, then our goal is to find k representative pixels so that pixels will have the same label as the nearest representative pixel. This can be done using the k-means algorithm, where we randomly initialize the k representative pixels, then iteratively look for the ideal representative pixels by minimizing the total distances to the current representative pixels. There are also variations that involve adding a heuristic in how we initialize the representative pixels [11], but the main idea remains the same.

This method is intuitive and can form an interpretable labelling, but its disadvantage is in finding a good representation of distance. This is because the importance of different features cannot be inherently discovered in clustering methods, so the distance must be a pre-determined function of the pixel features [12]. Additionally, the representative pixel has the same number of features as the original pixels, so the number of dimensions in the mapping from pixels to labels is limited by the number of features in the original pixels.

2.2. Neural Network Models

Neural network models have shown to produce better results in computer vision than conventional methods [3]. As our model is based on U-Net, a neural network model, the models to be discussed in this subsection will be similar in learning capabilities. However, the main difference will be in architecture, as U-Net has a distinct architecture, which we then augment with additional modules, such as the attention mechanism.

2.2.1 Convolutional Networks

Convolutional networks are usually used for image classification, so that an image will output a single class label instead of a label for each pixel. However, one paper implemented a convolutional network model by using a sliding window that uses local regions to output a label for each pixel [13].

This model increases the amount of training data as each image has several local regions that serve as input for training. However, this means that each local region is separately trained, so there is some redundancy that causes the network to train slower. The improvement of U-Net to this model is that we first down-sample the image, just as in normal convolutional networks, to reduce redundancy and produce an encoded image. We then up-sample the encoded image in later layers after learning the finer contextual features.

2.2.2 DeepLab

The DeepLab model takes a different approach as U-Net in applying shared convolutional filters for image segmentation. Instead of down-sampling an image, which will later be up-sampled, DeepLab instead applies atrous filters [14], which means that the filters are up-sampled by introducing holes in the filter.

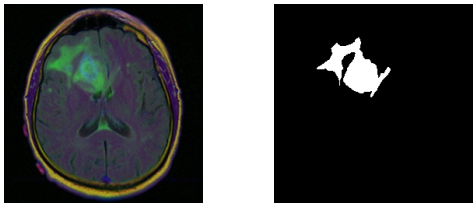
The DeepLab model allows for a higher resolution output image when compared to U-Net since the images were not up-sampled from lower resolutions. However, both models are considered state-of-the-art and we chose to implement only the U-Net model with different variations. This allows us to not only apply a state-of-the art model in tumor detection, but also to explore model variations with added modules that can prove useful in future research.

3. Dataset

The dataset used in this project is called the LGG Segmentation Dataset[1]. This dataset is published on the Kaggle website and aims to inspire more computer vision research in medical imaging.

It contains both the original brain MRI scan and the correct image segmentation, which was manually labelled by professional doctors. An example is seen in Figure 1, where white pixels indicate tumors while black pixels indicate non-tumors in Figure 1b. In total, there are 3929 pairs of images in the dataset, with each image having original size 512x512. The images are then separated based on the original patient, with each patient having multiple pairs of images in the dataset.

[1] www.kaggle.com/mateuszbuda/lgg-mri-segmentation



(a) Original MRI Scan (b) Image Segmentation

Figure 1: Sample RGB Image and Image Segmentation

3.1. Data Pre-processing

Before any training was done on the images, extensive data pre-processing steps were applied. These include cropping the images, resizing them to a 224x224 size square image, and then normalizing all pixel values channels wise. These steps were applied to both the original images and the ground-truth image segmentation mask to normalize the shape and colors of the images used. Additionally, for each pair of images, we also add a training sample that is a scaled version of the original image with factor f, and a training sample that is a rotation of the original image by an angle a degrees where f is drawn randomly from $[0.95, 1.05]$ for each image, and a is drawn randomly from $[-15, 15]$ for each image.

4. Problem Statement

Image segmentation is the task that relates to assigning a class or label to each pixel. We are given a labelled dataset consisting of pairs of RGB image and the desired image segmentation, with each pixel in the image segmentation as a label from the set $\{1, 2, ..., C\}$. We are tasked with finding a model that can accurately predict this labelling.

Specifically, our input consists of the RGB channels of an m by n image, and for each pixel j, we will output the predicted probability $\hat{y}_{i,j}$ for each class i such that for all pixels j, we have

$$\sum_{i=1}^{C} \hat{y}_{i,j} = 1 \tag{1}$$

We have $C = 2$ for the task, as we have one class for the pixels that contain tumors, and one class for the pixels that do not contain tumors.

4.1. Loss Function

Since we only have two pixel classes, the output of the model can be viewed as a prediction map, the size of which is the same as the input image, with every pixel in the map being the probability that the pixel is a tumor. Since we have the desired image segmentation for each image, one possible loss function we can use to train the model is categorical cross entropy (**CCE**), as we are essentially doing a pixel classification task. However, another option which we are currently trying is called **dice_loss**, which essentially measures the overlap between two samples. The formulation for both loss functions can be found below.

$$\textbf{CCE} = -\sum_{i=1}^{C}\sum_{j=1}^{N} y_{i,j} \log \hat{y}_{i,j} \tag{2}$$

$$\textbf{dice\_loss} = 1 - \frac{1}{C}\sum_{i=1}^{C} \frac{\sum_{j=1}^{N} 2y_{i,j}\hat{y}_{i,j} + \epsilon}{\sum_{j=1}^{N} y_{i,j} + \sum_{j=1}^{N} \hat{y}_{i,j} + \epsilon} \tag{3}$$

where i, j denote the class number and pixel number, $y_{i,j}, \hat{y}_{i,j}$ denote the actual and predicted value of the i^{th} class at the j^{th} pixel, and C, N denote the total number of classes and pixels respectively. ϵ is a small number added to avoid division by zero. Note that $y_{i,j}$ is either 0 or 1, but $\hat{y}_{i,j}$ can be any float from 0 to 1.

4.2. Evaluation

Aside from the two loss functions mentioned above, we will also use another metric to evaluate the results of the model. We will employ a commonly used metric known as **IOU** (Intersection over Union). As the name suggests, this metric measures how similar the predicted segmentation map is compared to that of the ground truth provided in the dataset.

$$\textbf{IOU}_{\text{image}} = \frac{\textbf{TP}}{\textbf{TP + FP + FN}} \tag{4}$$

$$\textbf{IOU}_{\text{mean}} = \frac{1}{M}\sum_{i=1}^{M} \textbf{IOU}_i \tag{5}$$

where **M** denotes the total number of images, **TP** denotes the true positives (we predicted a pixel to be that of a tumor), **FP** denotes the false positives (we predicted the pixel to be that of a tumor but it is not), and **FN** denotes false negatives (we did not predict the pixel to be part of a tumor but it is). We then compute the average of $\textbf{IOU}_{\text{image}}$ for all images in the validation dataset as our metric, $\textbf{IOU}_{\text{mean}}$.

5. Models

As mentioned in previous sections, we use the U-Net model architecture as the base model for the medical image segmentation task. As the U-Net model is already state-of-the-art, we expect to see good results. However, we also have models that explore either a recurrent structure or an attention mechanism to see whether these model variations can get better results. The description and architecture of each model variation will be discussed below.

5.1. U-Net

The base model architecture used is U-Net [2], named for its U-shaped architecture. It involves down-sampling an image using 3x3 convolution layers and 2x2 max-pooling layers, and then up-sampling the image using 3x3 convolution layers and 2x2 up-sampling layers.

For each down-sampling layer, we double the number of features from the previous layer, and for each up-sampling layer, we halve the number of features. Additionally, the number of down-sampling and up-sampling layers are equal, and the result from the n^{th} down-sampling layer will be concatenated with the result from the n^{th} up-sampling layer to produce the $n-1^{\text{th}}$ up-sampling layer, where we count layers from the top of the U shape as the first layer. The exact architecture is detailed in Figure 2.

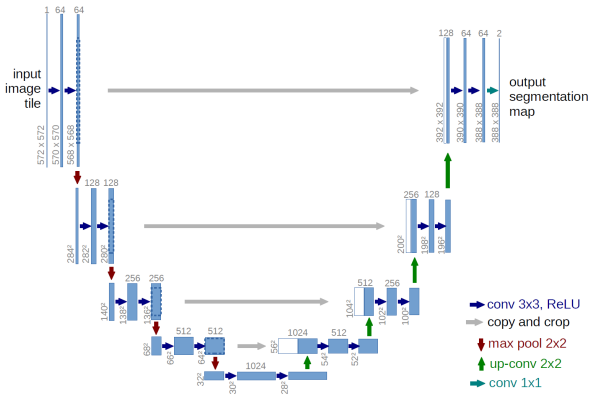


Figure 2: U-Net Model Architecture

Note that this model is able to learn both general and specific information since the up-sampled image contains contextual information while the image concatenated from the down-sampling layer captures location information. Additionally, this model can also train on an arbitrary size, since we down-sample and up-sample by a fixed factor instead of a fixed number.

5.2. RU-Net and R2U-Net

The RU-Net and R2U-Net models [15] draw inspiration from two different sets of models: (1) Recurrent Convolutional Neural Networks (RCNN) and Recurrent Residual Convolutional Neural Networks (RRCNN), which have shown superior performance in object recognition tasks, and (2) the previously discussed U-Net model. The overall architecture of the RU-Net model can be seen in Figure 3, whose architecture looks similar to the architecture shown in Figure 2.

RU-Net and R2U-Net have a similar architecture to U-Net since we still encode the data by down-sampling the original image and retrieve the prediction by up-sampling the encoded image. The difference lies in the layers used

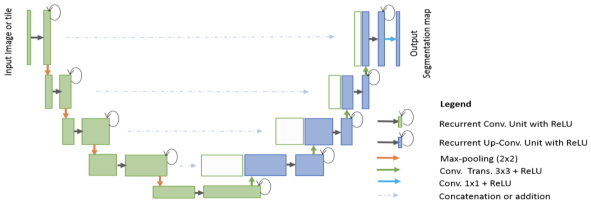


Figure 3: RU-Net Model Architecture

for the models. While U-Net uses convolutional layers to decode and encode the data, RU-Net and R2U-Net use recurrent convolutional layers (RCL). Meanwhile, the difference between RU-Net and R2U-Net is that R2U-Net uses residual units together with RCLs to address common problems found during the training of deep convolutional layers, such as vanishing gradients. This difference in the architectural units used in U-Net, RU-Net and R2U-Net is outlined in Figure 4.

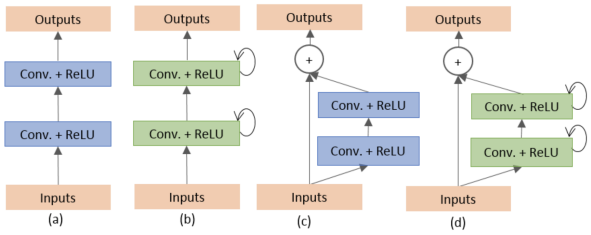


Figure 4: Variations of architectural units used in (a) U-Net, (b) RU-Net (c) Residual U-Net, and (d) R2U-Net.

The idea of this variation is that introducing a recurrent or residual layer will help the model train better and increase its learning capabilities to better capture the mapping function from the original image to the segmented image.

5.3. Attention U-Net

Another state-of-the-art U-Net variant is called the Attention U-Net [16]. Like its name suggests, it uses an attention gating mechanism on top of the standard U-Net architecture to aid in learning. This is a method commonly used in natural language processing tasks such as language modelling and language translation to produce state-of-the-art results. In natural language processing, attention is applied to words, but in this model, attention is applied to the down-sampled representations of the data. This has the advantage of allowing the model to be more sensitive to foreground pixels without requiring any complicated heuristic. Furthermore, this does not require training any new parameters. The overall model architecture can be seen in Figure 5.

In the Attention U-Net model, the input image is progressively filtered and down-sampled by factor of 2, much like U-Net, at each step in the encoding section of the net-

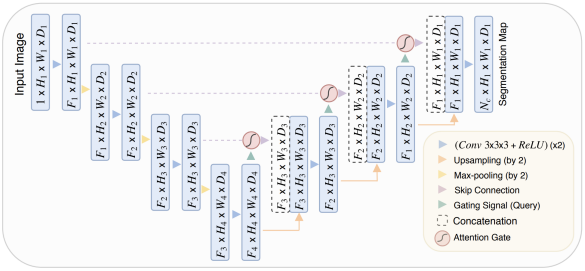


Figure 5: Attention U-Net Model Architecture

work. Attention Gates (AG) filter the features from the down-sampling layers that will later be concatenated with the up-sampled layers. An example of the schematic of AG is shown in Figure 6.

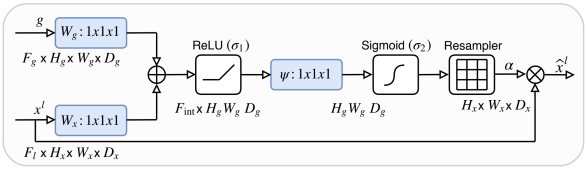


Figure 6: Schematic of the additive attention gate (AG)

In the figure, the input feature, denoted by x^l, is multiplied by an attention coefficient, denoted by α, that is computed in the AG. This computation for α in the AG is achieved by using the contextual information from coarser scales as gates.

5.4. Attention R2U-Net

This model is a combination of the unique model architecture designs of the R2U-Net and Attention U-Net models to see whether there is any synergy between the two types of models. It stands to reason that different modules that make each model unique and successful can yield even better results when combined together. The model architecture can be seen in Figure 7.

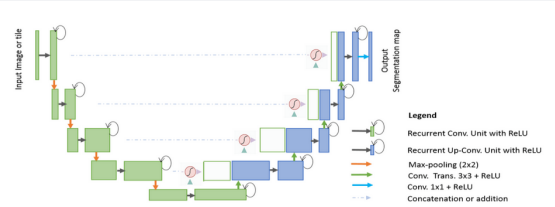


Figure 7: Attention R2U-Net Model Architecture

The Attention R2U-Net model essentially uses the same architecture units as the R2U-Net model, the RCL with residual units, and adds the same attention mechanism used in the Attention U-Net model. The advantage is that we are able to combine both features and see whether there are any significant changes in the results.

6. Experiments and Results

In this paper, we show the different experiments done on the U-Net model variations discussed in Section 5 and the results of each experiment.

6.1. Model Evaluation

We first test each model variation to see which one will have the best performance. The trained models use standardized parameters, with a **batch_size** of 16 and a **learning_rate** of $1e{-}4$ for 50 **epochs**. We then have 5 **layers** each for up-sampling and down-sampling, with $64, 128, 256, 512,$ and 1024 features respectively. Each convolution layer has a **stride** of 2 and **kernel_size** of 2, and is followed by a **ReLU** activation layer. The results for the model are shown in Table 1.

Model	**dice_loss**	**IOU**$_{\text{mean}}$
U-Net	0.866597	0.779079
R2U-Net	0.854132	0.757487
Attention U-Net	**0.875512**	**0.788855**
Attention R2U-Net	0.706535	0.562255

Table 1: Model Results

After performing the experiments, we can see in Table 1 that the Attention U-Net model performs the best with the given dataset. The original U-Net model does come in a close second, with the R2U-Net model coming in third. An interesting thing to notice here is that combined Attention R2U-Net variant performs the worst and by a substantial margin. This suggests that the resulting model is too complicated for the given dataset, resulting in some overfitting. We can see an interesting example of this failed learning in Figure 8.

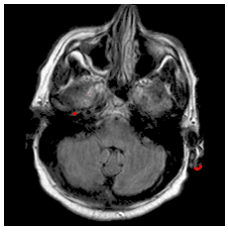
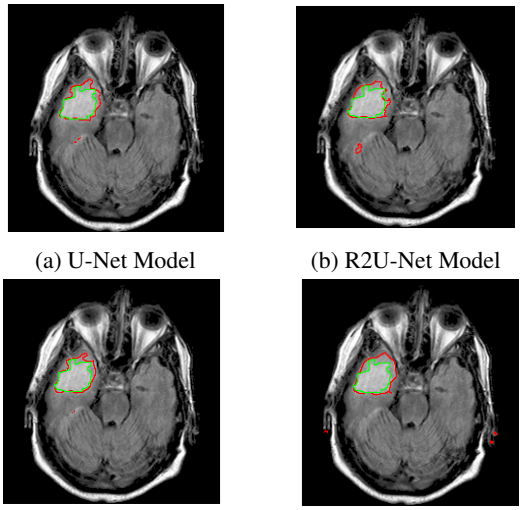


Figure 8: Failed Learning of Attention R2U-Net Model

There are no green regions in the image as there is no tumor in the ground-truth image. All the three other models predicted this correctly but the Attention R2U-Net model fails in this example.

Additionally, an example of the results for all 4 models can be found in Figure 9. The models seem to be able to generally identify the shape and location of the tumors, but there are some false positives in all models, with the Attention U-Net having the least significant error.



(a) U-Net Model (b) R2U-Net Model

(c) Attention U-Net Model (d) Attention R2U-Net Model

Figure 9: Prediction of Different Models

The red outlines are the predicted region of tumor while the green outline is the actual region of tumor.

6.2. Hyperparameter Tuning

In this section, we explore different values of the hyperparameters to see whether they have a significant impact to the results of the experiment. We will only be experimenting with the Attention U-Net model for this sub-section as it was capable of producing the best results, and testing on other similar models will have redundant results.

6.2.1 Number of Layers

For this parameter, we try to see whether changing the number of layers in the U-Net base architecture will lead to a significant change in results. The results are outlined in Table 2. For clarification, if only 3 layers are used, then these will have $64, 128$, and 256 features respectively. Meanwhile, if 4 layers are used, then these will have $64, 128, 256$, and 512 features respectively.

The results in Table 2 indicate that increasing the number of layers does help with the overall model performance, as the original model with 5 layers has the best results. This means that by adding a new layer, we can get a better encoded image, which is later able to produce better attention weights and an overall better result.

# of Layers	**dice_loss**	**IOU$_{\text{mean}}$**
3	0.816272	0.712297
4	0.870918	0.785391
5	**0.875512**	**0.788855**

Table 2: Performance of different number of layers

6.2.2 Batch Size

We also tried to vary the batch size to see whether this will have an affect on the final results. However, the results shown in Table 3 indicate that the batch size does not have any significant effect to the model performance. But for optimization purposes, it is worth noting that 16 seems to be the best batch size for this dataset.

Batch Size	**dice_loss**	**IOU$_{\text{mean}}$**
8	0.873569	0.787258
16	**0.875512**	**0.788855**
24	0.873112	0.785030

Table 3: Performance of different batch sizes

6.2.3 Learning Rate

We also checked whether the learning rate parameter will have a significant effect on the results. This is because the learning rate can greatly affect whether the model can converge properly and how long the model will take to train.

Learning Rate	**dice_loss**	**IOU$_{\text{mean}}$**
$1e{-3}$	0.865332	0.774038
$1e{-4}$	**0.875512**	**0.788855**
$1e{-5}$	0.832791	0.733287

Table 4: Performance of different learning rates

The results in Table 4 indicate that the learning rate of $1e{-3}$ is too large, as it cannot reach the performance of the original model that had a learning rate of $1e{-4}$. We also have worse results by using a learning rate of $1e{-5}$ with 50 epochs, indicating that the learning rate is too small and more epochs would be needed to match the results of the original model.

6.2.4 Optimizer

Lastly, we also checked whether the results of the model would be better if we used a different optimizer. This is because the optimizer, similar to the learning rate, can also af-

fect whether the model will converge properly and how long the model will take to train. For the experiment, we considered using another popular optimizer called RMSProp optimizer. However, the results in Table 5 indicate that the Adam optimizer still performs better than the RMSProp optimizer.

Optimizers	**dice_loss**	**IOU**$_{\text{mean}}$
RMSProp	0.874698	0.788030
Adam	**0.875512**	**0.788855**

Table 5: Performance of different optimizers

7. Conclusion

In this study, we applied a popular image segmentation model, U-Net, to the task of segmenting a brain MRI scan into tumor and non-tumor pixels. The results showed that the U-Net model is generally able to capture the location and size of tumors in the image. By adding an attention mechanism, the model is also able to improve its results, meaning that attention mechanisms can be applied not only to natural language processing tasks but also to medical image segmentation tasks. Additionally, we learned that increasing the number of layers in the U-Net base architecture can improve the results of the model, while finding the correct learning rate is also important to achieve better results.

8. Future Work

Medical imaging is a field of computer vision that should be emphasized more because of its potential to benefit the human population. This paper tested with using the U-Net model and attention mechanisms to get good results in medical image segmentation. Future research should be done on applying different models such as DeepLab in medical image segmentation. Additionally, as attention mechanisms proved effective in this paper, future research should also be done on identifying other ways to apply attention mechanisms in different medical imaging tasks.

References

[1] He, K., Zhang, X., Ren, S. and Sun, J., 2015. Deep Residual Learning for Image Recognition.

[2] Ronneberger, O., Fischer, P. and Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation.

[3] Voulodimos, A., Doulamis, N., Doulamis, A. and Protopapadakis, E., 2018. Deep Learning for Computer Vision: A Brief Review.

[4] Huda, W. and Abrahams, R.B., 2015. X-Ray-Based Medical Imaging and Resolution.

[5] Wang, Y., Huang, M. and Zhao, L., 2016. Attention-based LSTM for aspect-level sentiment classification.

[6] Senthilkumaran, N. and Vaithegi, S., 2016. Image Segmentation By Using Thresholding Techniques For Medical Images.

[7] Otsu, N., 1979. A threshold selection method from gray-level histograms.

[8] Batenburg, K.J. and Sijbers, J., 2009. Adaptive thresholding of tomograms by projection distance minimization.

[9] Bhargavi, K. and Jyothi, S., 2014. A Survey on Threshold Based Segmentation Technique in Image Processing.

[10] Lloyd, S.P., 1982. Least squares quantization in PCM.

[11] Arthur, D. and Vassilvitskii, S., 2007. k-means++: The Advantages of Careful Seeding.

[12] Loohach, R. and Garg, K., 2012. Effect of Distance Functions on K-Means Clustering Algorithm.

[13] Ciresan, D.C., Gambardella, L.M., Giusti, A. and Schmidhuber, J., 2012. Deep neural networks segment neuronal membranes in electron microscopy images.

[14] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A., 2017. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs.

[15] Alom, M.Z., Hasan, M., Yakopcic, C., Taha, T.M. and Asari, V.K., 2018. Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation.

[16] Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B., Glocker, B. and Rueckert, D., 2018. Attention U-Net: Learning Where to Look for the Pancreas.