2019 22nd International Conference on Computer and Information Technology (ICCIT), 18-20 December 2019

3D U-Net: Fully Convolutional Neural Network for Automatic Brain Tumor Segmentation

Syed Fahim Ahmed
dept. of CSE
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
fahimdbolt@gmail.com

Fairuz Shezuti Rahman
dept. of CSE
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
rodoshi_1610@yahoo.com

Tasmia Tabassum
dept. of CSE
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
tasmiaarony@gmail.com

Md. Tariqul Islam Bhuiyan
dept. of CSE
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
tariqul.rizvi@gmail.com

***Abstract*—A principal problem in brain tumor-related diagnosis, monitoring and treatment is the assessment of the tumor area and location. Automatic segmentation is attractive in this context, as it grants us a faster approach and potentially more accurate description of tumor parameters. In this paper, we have proposed a semantic segmentation technique to 3D MRI images based on popular U-net architecture which is a Fully Convolutional Network (FCN) and robust algorithm. Our presented network is evaluated on the Multimodal Brain Tumor Image Segmentation (BRATS 2015) dataset, which contains 274 people's brain MRI images. We have used soft dice loss function to subsist with class disparities and augmented the data to prevent overfitting. Cross-validation of tumor segmentation has also displayed that our model architecture can get promising segmentation efficiently than other popular architectures with a dice score of 0.79 for the whole tumor.**

***Index Terms*—3D U-net, CNN, Brain Tumor, Segmentation, Deep Learning**

I. Introduction

The brain tumor is one of the most dangerous and vulnerable diseases nowadays. Medical science is approaching with every perspective for the variable and constant result for this issue. In the case of tumor affected diseases, adults mostly suffer from primary central nervous system lymphomas and gliomas, which turns into malignant almost 80% cases [1]. For the diagnosis of the brain tumor, multimodal magnetic resonance imaging (MRI) are normally used which provide crucial complementary information. Formerly, the treatment process was primer and manual where the doctors would have to construct a dreadful operation based on their prior knowledge. Not only the process was expensive, but also the successive rate was quite low. Recently, computer-aided techniques have been developed in a great number for analyzing and visualizing magnetic resonance images (MRI). Using these techniques, many researchers have detected and segmented the brain tumor which is faster and accurate than manual processes.

Previously, machine learning algorithms like SVM, KNN, random forest classifier, clustering algorithms, etc. were used in the field of brain tumor detection and segmentation. But now, convolutional neural network (CNN) based methods are widely employed in brain tumor segmentation as it works better on image data than other algorithms. The convolution architecture that has accustomed to this paper for segmentation is called 3D U-Net which was introduced by Ronneberger et al. [2] for biomedical images.

II. Related Work

The number of publications devoted to automated brain tumor segmentation has grown exponentially in the last several decades. CNN works more efficiently than other methods in the case of brain tumor segmentation. Sergio Pereira et al. [3] have used deep architectures with small convolutional kernels for the segmentation of gliomas in MRI images. The proposed architecture obtained DSC scores of 0.78, 0.65 and 0.75 for complete, core and enhanced regions, respectively in the BRATS 2015 dataset. Tustison's method takes 100 min to compute predictions per brain as reported in Menze et al. (2014) [4], while the InputCascadeCNN proposed in Mohammad Havaei et al. [5] takes 3 min by using fully convolutional architecture and the GPU implementation, which is over 30 times faster. Javeria Amin et al. [6] achieved 99.8% DSC on Flair, 100% results on DWI, 98.0% on T2, 97.4% on T1 and 95.4% on T1-contrast modalities by building a 7 layered CNN architecture including 03 convolutional, 03 ReLU and a softmax layer. Ali Isin et al. [7] presented reviews of the forefront methods based on deep learning and a brief overview of traditional techniques.

Long et al. [8] proposed an architecture that combined the high-level representation from deep decoding layers and shallow encoding layers produces detailed segmentation. This method has demonstrated promising results on natural images [9] and can also be applied to the biomedical images [10]. Ronneberger et al. [2] introduced the U-Net, which employed the skip-architecture, by solving the cell tracking problem. Hao Dong et al. [11] have narrated down flaws of

978-1-7281-5842-6/19/$31.00 ©

the traditional way of segmentation of brain tumors from MRI images and using the BRATS 2015 dataset they have formatted a new approach 2D fully convoluted segmentation network for overcoming those frailties.

El-Sayed A. El-Dahshan et al. [12] have used feedback pulse-coupled neural network (FPCNN), discrete wavelet transform, principal component analysis as well as back-propagation neural network (BPNN) in several stages of their architecture. Similarly, Kailash D.Kharat et al. [13] proposed both Feedforward artificial neural network (FF-ANN) and Back Propagation Neural Network (BP-ANN) for the classification of the magnetic resonance human brain images. Dina Aboul Dahab et al. [14] used a modified version of the conventional PNN method helps to classify the brain tumor. Similarly, Pauline John [15] proposed the PNN algorithm for segmentation.

Eman Abdel-Maksoud et al. [16] used the K-means clustering technique, which is very popular in image processing, integrated with the Fuzzy-C-means algorithm. Although the K-mean algorithm can detect a brain tumor faster than Fuzzy C-means. In contrast, Matthew C. Clark et al. [17] and Stefan Bauer et al. [18] used unsupervised clustering algorithms for brain tumor segmentation. B. Devkota et al. [19] established a new algorithm, Spatial Fuzzy C-Means (SFCM) by another method of high peak signal to noise ratio (PSNR) for segmentation and achieved the accuracy of 92%. Stefan Bauer et al.'s [20] segmented the brain tissue using a soft margin SVM classifier which is based on the LibSVM implementation. Regularization is done in two different stages using a Conditional Random Field method. The gross tumor volume for intra-patient regularized and inter-patient regularized are 0.84 and 0.77 respectively. Discrete wavelet-based GA (genetic algorithm) was implemented in [21] to detect tumors from the MRI brain images. Following the process of the optimal or sub-optimal data partition, they have implemented k-means unsupervised clustering methods into genetic algorithms.

III. Methodology

In our research, we have done tumor segmentation by segmenting the parts of the image with similar values. We separated the tumor from the background of brain and show the location of that tumor. Figure 1 shows the flowchart of our overall work.

A. Pre-processing

To segment the object with detailed information, pre-processing is an essential section that directly influences the model architecture. Before segmentation, pre-processing was executed on all the MRI images except for the ground truth images. We used standardization on the data so that, the images were conditioned in a manner that no lower values

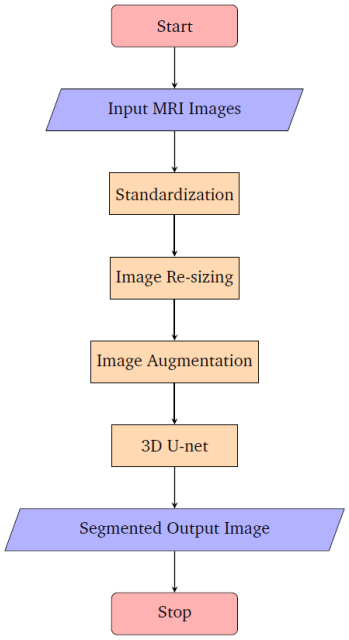

Fig. 1. Flowchart of Methodology

can be dominated by the higher values. For standardization, we applied,

$$X_{stand} = \frac{x - mean(x)}{standardDeviation(x)} \quad (1)$$

1) Re-sizing Image: Our original data is very huge in term of size. So, for effieciency, we re-sized the images. First, we re-size all 4 kinds of the 3D MRI images from $240 \times 240 \times 155$ to a $32 \times 32 \times 32$ uniform size. After that, we concatenate all 4 kinds of images (T1, T2, T1c and Flair) into numerical arrays for all 274 patients (220 HGG and 54 LGG).

2) Image Augmentation: The image augmentation process creates more data from the original one which improves the network performance. For segmenting our dataset we applied a set of data augmentation techniques showed in Table I. The augmentation methods were programmed using Keras deep learning library.

TABLE I
Applied Data Augmentation Methods

Methods	Range
Flip horizontally	50% probability
Flip vertically	50% probability
Rotation	$\pm 20^{\circ}$
Shift	10% on both horizontal and vertical direction
Shear	20% on horizontal direction
Zoom	$\pm 10\%$
Brightness	$\gamma = 0.8 \sim 1.2$

B. Semantic Segmentation

To analyze the input image easily, semantic segmentation is used in our model. In this process, the images are partitioned into different regions. Every pixel in the image belongs to a

particular class and the similar pixels are all assigned to a single color which segments the image quite efficiently.
One of the popular approaches for image segmentation models is to maintain an encoder/decoder structure where one have to down-sample the spatial resolution of the input, developing lower-resolution feature mappings and the up-sample of the feature representations into a full-resolution segmentation map. The approach of using a "fully convolutional" network trained end-to-end and pixels-to-pixels for image segmentation was introduced by Long et al. [8] in 2014. By using this encoder-decoder model architecture, the U-net method was built. U-net architecture basically only takes important information from the images and make into smaller part in the encoding module then padding with similar information and make large the smallest convoluted information in decoding module. Lastly, the output image is the same size as the input image but with significant changes which are semantic segmented images.

C. 3D U-net Based Deep Convolutional Networks

U-net was developed by Olaf Ronneberger et al. [2], based on deep level encoding and decoding architecture for biomedical image segmentation. The model has two paths. The first path is the construction path which is also named as the encoder part that is used to capture the context in the image. The encoder is just a traditional stack of convolution layers as well as max-pooling layers. The second path is the symmetric expanding path which is also named as the decoder. It is implemented to enable precise localization using transposing convolution layers. Thus it is an end-to-end Fully Convolutional Network(FCN) which only have convolutional layers and does not have any dense layer as it can accept an image of any size. The architecture is a symmetric U-shape and that is why it is named U-net architecture. Before we dive into our model 3D U-net, it is important to understand the different operations that are typically used in a fully convolutional network.

1) Convolution Operation: In convolution operation, we produce feature maps from input 3D image. There are three inputs to this convolutional operation.

- **Input Size:** A 4D volume (input image) with the size of $D_1 \times H_1 \times W_1 \times C_1$, where D_1 is the depth, H_1 is the height, W_1 is the width and C_1 is the color channel of the input image.
- **Filters:** The dimensionality of the output space that means if there is 'k' kernels or filters the number of output will increase to 'k' from unit input size.
- **Kernel Size:** $D_2 \times H_2 \times W_2$, specifying the depth, height, and width of the 3D convolution window.

Output Size: A 4D volume which is also called an output image or feature map with the size of $D_3 \times H_3 \times W_3 \times C_3$. One important term used frequently is the Receptive field (context), which is the area of the input image that the filter covers at any given point of time.

2) Max Pooling Operation: The function of pooling is to reduce the size of the feature detector so that, we have fewer parameters in the network architecture. Basically from every $2 \times 2 \times 2$ block of the input feature detector, we select the maximum pixel value and thus obtain a max pooled feature map. Both convolution operation and especially the pooling operation reduce the size of the image which is called as down-sampling. All the parameters for this operation is given below.

- **Input Size:** A 5D tensor with the shape of $B_1 \times D_1 \times H_1 \times W_1 \times C_1$.
- **Pool Size:** $D_2 \times H_2 \times W_2$, the mask size for pooling.
- **Strides:** $D_3 \times H_3 \times W_3$, a tuple of three integers which defines how much the mask has to leap for each step.

Output Size: A 5D volume which is also called an output image of $B_4 \times D_4 \times H_4 \times W_4 \times C_4$.

3) Up-sampling Operation: The output of semantic segmentation are complete high-resolution images in which all the pixels are classified. Thus if we use a regular convolutional network with pooling layers and dense layers, we will lose some information. To make segmentation more meaningful, there is a need to up-sample the image, i.e. convert a low-resolution image into a high-resolution image to regain the loss information.

- **Input Size:** A 5D tensor with the shape of $B_1 \times D_1 \times H_1 \times W_1 \times C_1$.
- **Size:** $D_2 \times H_2 \times W_2$, The up-sampling factors for depth, height and width which means how much it going to up sample the data.

Output Size: A 5D volume which is also called an output image of $B_3 \times D_3 \times H_3 \times W_3 \times C_3$.

4) Transposed Convolution Operation: Transposed convolution, which is sometimes also called as deconvolution, is a technique to perform up-sampling of an image with learnable parameters. Transposed convolution is exactly the opposite process of a normal convolution operation which practically means the input volume which is a lower resolution image converted into a higher resolution image.

5) Activation Function: Activation function helps CNN based model to learn new data by activating the neuron related to the information given with high weight and it ignores the rest, we have used ReLU (Rectified Linear Unit) Activation Function in this model. Equation used by ReLU function is,

$$\varnothing(x) = max(x, 0) \tag{2}$$

We used the ReLU in all normal convolution operation as well as in all transposed convolution operation except for the last one. The last transposed operation was done with the help of the sigmoid function to get the main output semantic segmented image.

D. Dataset

We have used BRATS 2015 dataset in our project where we have 220 HGG or High Grade Glioma patients' data and 54 LGG or Low Grade Glioma patients' data for training. Also,

110 data for testing with both HGG and LGG type are given. All of the MRI images are of 4 types: Flair, T1, T1c and T2. We have taken 80% MRI images as our training data, 20% MRI images as our validation testing data from BRATS 2015 training dataset.

IV. Experimental Setup

A. Model Architecture

Our network is inspired by the U-Net architecture of Olaf Ronneberger et al. [2] who also made the architecture for biomedical images. Like [8], this network architecture uses the skip-architecture that combined the down-sampling (encoding) path and an up-sampling (decoding) path as shown in figure 2. Our model has 5 Convolutional blocks for down-sampling

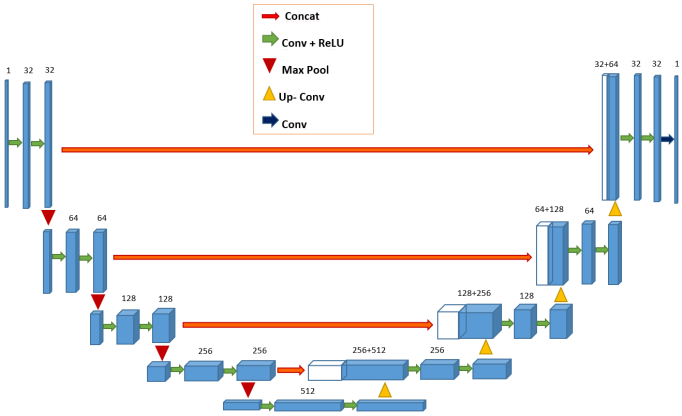

Fig. 2. U-Net Architecture of our Model

path and each block has two convolutional layers with a filter size of 3×3, a stride of 1 in all three directions and ReLU function as the activation function which increases the number of feature detectors from 1 to 512. Max pooling is used in the end of every block with stride 1 with a pool size of $2 \times 2 \times 2$ in down-sampling. The size of the feature detectors decreases from $32 \times 32 \times 32$ to $2 \times 2 \times 2$. In the up-sampling path, each level block starts with filter size of $2 \times 2 \times 2$ which doubles the size of the feature detectors in three dimensions but decreases the number of feature maps by two, so the size of the feature detectors increases from $2 \times 2 \times 2$ to $32 \times 32 \times 32$. Two convolutional layers reduce the number of the feature detectors of the concatenation of up-sampling (deconvolution) feature detectors and the feature maps from the encoding path. At last, a $1 \times 1 \times 1$ convolutional layer is built to reduce the number of the feature detectors to two that reflect the foreground and background segmentation respectively.

1) Training and Optimization: During the training process, the Soft Dice metric described in [9] was used to evaluate our network model's performance as a cost function. Soft Dice can be considered as a differentiable form of the original Dice Similarity Coefficient (DSC) as we get it from subtracting the obtained DSC from 1 [9]. Training deep neural networks requires stochastic gradient-based optimization which minimizes the cost function concerning its parameters and that is why we implemented the adaptive moment estimator (Adam) [22] to estimate the parameters. In general, Adam employs the first and second moments of gradients for updating and correcting the moving average of the current gradients. The parameters of our Adam optimizer were set as learning rate = 0.0001 and the maximum number of epochs = 300. All weights were assigned by normal distribution with a mean of 0 and a standard deviation of 0.01, and all biases were assigned as 0.

2) Performance Evaluation: The evaluation of our network has been done on the complete tumor region of the brains using a five-fold cross-validation method for the HGG and LGG data, respectively. The segmentation have been evaluated using the dice soft coefficient which provides the overlap measurement between the brain tumor and the segmentation results of our fully automatic method that is,

$$DSC = \frac{2TP}{FP + 2TP + FN} \tag{3}$$

In this equation, the true positive, false positive and false negative are denoted as TP, FP, FN measurements, respectively. Another performance metrics, accuracy is used to evaluate the whole data and that is,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

where, TP = True positive; FP = False positive; TN = True negative; FN = False negative

V. Experimental Results & Discussion

A. Performance measure of the Model

We used Dice soft coefficient for performance measurement. The loss function which is measured from Dice Soft Loss (DSL), is calculated by subtracting dice soft coefficient from whole probability.

TABLE II
Accuracy & Loss Function value for Tumor Segmentation

Data Type	Accuracy	Loss Function
Training	93.08%	19%
Validation Testing	92.09%	21%

From our model, we achieved 93.08% accuracy on training data and 92.09% accuracy on five-fold cross-validation testing. The loss function's value is also low with 19% in training data and 21% in validation testing data. Table II shows the measurement of this data.

1) Accuracy Curve for Validation Testing: The accuracy curve of the segmentation model is given in figure 3 which shows the accuracy for training and validation testing data. On 1st epoch, the accuracy was 0.9. Moving epoch by epoch, the accuracy increases for both training and validation testing data. After our final epoch which is 300, the accuracy for training data is 0.93 and for validation testing data is 0.92.

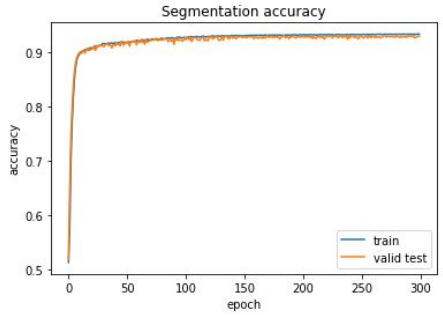

Fig. 3. Accuracy graph

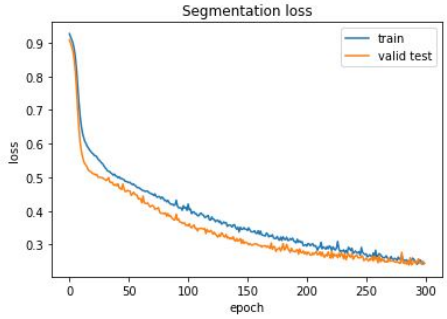

Fig. 4. Loss Function graph

2) Loss Function Curve for Validation Testing: The result that we obtained from the segmentation loss function is measurement by dice soft loss. Figure 4 is the visual representation of loss function.

At the 1st epoch, the loss value was high for both data which was 92%. After increasing number of epoch, the loss value is decreased. The gradually downwards curves shows the efficiency of our model. Lastly at the 300th epoch the loss function's value for training data is 19% and for validation testing is 21%. If we increase the epoch number further, the value for loss function will decrease fractionally.

B. Segmentation Result for BRATS 2015

1) Five-fold Cross-Validation Result: In Flair images, using a long inversion time and a long effective echo time, the signal is made null from bright cerebrospinal fluid (CSF) of brain. High CSF is suppressed by long inversion time and the small periventricular lesions are visualized easily in the image.

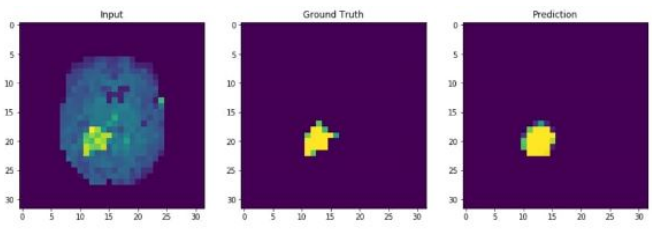

Fig. 5. Segmentation of Flair MRI for Patient-1

Figure 5 shows the segmented image of patient-1 which have been found by validation testing.

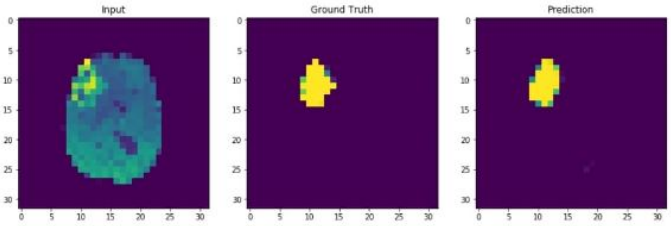

Fig. 6. Segmentation of Flair MRI for Patient-2

Figure 6 shows the segmented image of patient-2. This segmented images are obtained from model when five-fold cross validation is done. Our research model only highlights the tumor area and all the other pixels are colored as background.

2) Testing Result: The result of testing is found using the testing dataset of BRATS 2015. Testing data is feed to our trained model after five-fold cross-validation testing.

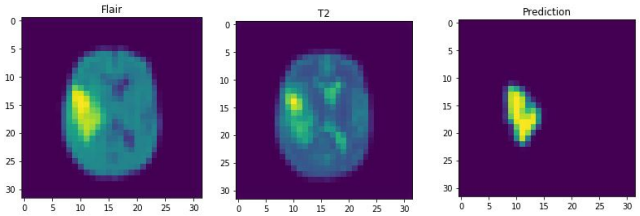

Fig. 7. Segmentation of Testing Data

In Figure 7, we have shown a random patient's Flair, T2 and predicted segmented MRI image respectively. The predicted segmented image gives almost accurate tumor size and location.

C. Comparison of Different Methods of Tumor Segmentation

We have compared our work's performance with three other papers who are the best papers in this field.
Sergio Pereira et al. [3] and Mohammad Havaei et al. [5] has achieved the DSC value of 0.78 but our DSC value is 0.79 in cross validation testing. Table III shows the comparison between [3], [5] and ours.

TABLE III
COMPARISON OF DIFFERENT METHODS IN RESPECT OF DSC

Method	Data	DSC
Proposed Method [figure 2]	BRATS 2015	0.79
Sergio Pereira et al. [3]	BRATS 2015	0.78
Mohammad Havaei et al. [5]	BRATS 2015	0.78

We used small 3D MRI patches with $32 \times 32 \times 32$ resolution to reduce the training time. For that reason, [8] took 22 hours 5 minutes for training the data and others took above one day but we trained it in only 12 hours 30 minutes.

In our proposed method, we have given the prediction in less than 1 sec. It is the lowest time consuming model. Sergio Pereira et al. [3] and Mohammad Havaei et al. [5] have given

the prediction in 8 min and $25sec \sim 3min$ respectively which is not as good as our model. Even the lowest time of Hao Dong et al. [11] is $2 \sim 3sec$ which is also higher than our time delay. In Table IV, comparison in respect of time is shown.

TABLE IV
COMPARISON OF DIFFERENT METHODS IN RESPECT OF TIME FOR PREDICTION

Method	Data	Time
Proposed Method [fig. 2]	BRATS 2015	less than $1sec$
Hao Dong et al. [11]	BRATS 2015	$2 \sim 3sec$
Sergio Pereira et al. [3]	BRATS 2015	$8min$
Mohammad Havaei et al. [5]	BRATS 2015	$25sec \sim 3min$

Our result can be increased in accuracy and efficiency, if we have used 12 GB GPU like [5], [3] and [11] on same dataset. Still the efficiency measurement of our model is highest.

VI. CONCLUSION

In this paper, we presented a segmentation technique using 3D U-Net inspired deep convolutional neural network architecture which was trained using only the provided training data, extensive data augmentation and a dice loss formulation. We achieved a mean dice score of 0.79 for the whole tumor. The training time was about 12 hours 30 minutes long as we only trained on 4 GB GPU. Due to time restrictions, we were limited in the number of architectural variants and data augmentation methods we could explore and resized the images into small $32 \times 32 \times 32$ patches with a batch size of 8. Thus the computation time for prediction has been decremented to handsome extent. Training with larger batch sizes and more convolutional filters in a multi-GPU setup should yield further improvements, especially provided that we did not observe significant overfitting in our experiments.

REFERENCES

[1] Judith A Schwartzbaum, James L Fisher, Kenneth D Aldape and Margaret Wrensch, "Epidemiology and molecular pathology of glioma," in Nature Clinical Practice Neurology volume 2, pp. 494-503 2006.

[2] Olaf Ronneberger,Philipp Fischer,Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015, pp. 234–241, 2015.

[3] Sergio Pereira, Adriano Pinto, Victor Alves, and Carlos A. Silva, "Brain tumor segmentation using convolutional neural networks in MRI Images," in IEEE TRANSACTIONS ON MEDICAL IMAGING: Volume 35, Number 5, 2016, pp. 1240–1251.

[4] Bjoern H. Menze et al. "The multimodal brain tumor image segmentation benchmark (BRATS)," in IEEE TRANSACTIONS ON MEDICAL IMAGING: Volume 34, Number 10, 2015, pp. 1993–2024.

[5] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoina, Hugo Larochelle, "Brain tumor segmentation with deep neural networks," in Medical Image Analysis: Volume 35, 2017, pp. 18–31.

[6] Javeria Amin, Muhammad Sharif, Mussarat Yasmin, Steven Lawrence Fernandes, "Big data analysis for brain tumor detection: Deep convolutional neural networks," in Future Generation Computer Systems: Volume 87, pp. 290–297, 2018.

[7] Ali Isin, Cem Direkoglu, Melike Sah, "Review of MRI-based brain tumor image segmentation using deep learning methods," in Procedia Computer Science: Volume 102, pp. 317–324, 2016.

[8] Jonathan Long, Evan Shelhamer, Trevor Darrell, " Fully convolutional networks for semantic segmentation," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431–3440, 2015.

[9] Fausto Milletari, Nassir Navab, Seyed-Ahmad Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in Fourth International Conference on 3D Vision (3DV), pp. 1–11, 2016.

[10] Michal Drozdzal, Eugene Vorontsov, Gabriel Chartrand, Samuel Kadoury and Chris Pal, "The importance of skip connections in biomedical image segmentation," in Deep Learning and Data Labeling for Medical Applications, pp. 179–187, 2016.

[11] Hao Dong, Guang Yang, Fangde Liu, Yuanhan Mo, Yike Guo, "Automatic brain tumor detection and segmentation using U-Net based fully convolutional networks," in Medical Image Understanding and Analysis, pp. 506–517, 2017.

[12] El-Sayed A. El-Dahshan, Heba M. Mohsen, Kenneth Revett, Abdel-Badeeh M. Salem, "Computer-aided diagnosis of human brain tumor through MRI: A survey and a new algorithm," in Expert Systems with Applications: Volume 41, Issue 11, 2014, pp. 5526–5545.

[13] Kailash D.Kharat, Pradyumna P.Kulkarni, M.B.Nagori, "Brain tumor classification using neural network based methods," in International Journal of Computer Science and Informatics, ISSN (PRINT) 2231–5292: Volume 1, Issue 2, 2012.

[14] Dina Aboul Dahab, Samy S. A. Ghoniemy, Gamal M. Selim, "Automated brain tumor detection and identification using image processing and probabilistic neural network techniques," in International Journal of Image Processing and Visual Communication, ISSN (Online) 2319–1724: Volume 1, Issue 2, 2012.

[15] Pauline John, "Brain tumor classification using wavelet and texture based neural network," in International Journal of Scientific & Engineering Research, ISSN 2229–5518: Volume 3, Issue 10, 2012.

[16] EmanAbdel-Maksoud, MohammedElmogy, RashidAl-Awadi, "Brain tumor segmentation based on a hybrid clustering technique," in Egyptian Informatics Journa: Volume 16, Issue 1, pp. 71–81, 2015.

[17] Matthew C. Clark et al., "Automatic tumor segmentation using knowledge-based techniques," in IEEE TRANSACTIONS ON MEDICAL IMAGING: Volume 17, Issue 2, pp. 187201, 1998.

[18] Stefan Bauer, Roland Wiest, Lutz-P. Nolte, and Mauricio Reyes, "A survey of MRI-based medical image analysis for brain tumor studies," in Physics in Medicine & Biology: Volume 58, Number 13, 2013.

[19] B. Devkota, Abeer Alsadoona, P.W.C. Prasad, A. K. Singh, A. Elchouemi, "Image segmentation for early stage brain tumor detection using mathematical morphological reconstruction," in Procedia Computer Science: Volume 125, pp. 115–123, 2018.

[20] Stefan Bauer, Lutz-P. Nolte, and Mauricio Reyes, "Fully automatic segmentation of brain tumor images using support vector machine classification in combination with hierarchical conditional random field regularization," in Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011, pp. 354–361.

[21] G Rajesh Chandra, Dr.Kolasani Ramchand H Rao, "Tumor detection in brain using genetic algorithm," in Procedia Computer Science: Volume 79, pp. 449–457, 2016.

[22] Diederik P. Kingma, Jimmy Ba, "Adam: A method for stochastic optimization," in 3rd International Conference for Learning Representations, San Diego, 2015.