

Brain Tumor Segmentation with Efficient and Low-Complex Architecture Using RCNN and Modified U-Net

Ananta Raha^(✉), Farjana Parvin, and Tasmia Jannat

Department of Computer Science and Engineering,
Rajshahi University of Engineering & Technology, Rajshahi, Bangladesh
ananta.raha.99@gmail.com^(✉), farjana@cse.ruet.ac.bd,
jannat22tasmia@cse.ruet.ac.bd

Abstract. In medical applications, the boundless potential of image processing utilizing Deep Neural Networks has grabbed the interest of researchers. Brain tumor segmentation, which is a crucial piece of task, determines the location and extent of tumor areas. Numerous techniques for segmentation have been suggested by researchers. One significant disadvantage of the existing architectures is the presence of a large number of trainable parameters. It makes the system complex, expensive to train, and unsuitable for integration in low-powered devices. In this paper, we present an efficient, two-stage approach for the effective segmentation of brain tumor from MRI images using RCNN and a modified U-Net. The proposed system was evaluated and verified using a publicly available Figshare dataset [1]. The system is low-complex with small number of parameters compared to other existing architectures. It was tested and compared to the original U-Net, and despite having a large decrease in total trainable parameters, it obtained a comparable performance with an accuracy of 99.78%, IoU of 89.76% and a dice score of 94.53% in our experiments.

Keywords: Complexity reduction, Brain tumor, Segmentation, Low-complex, RCNN, U-Net

1 Introduction

Machine learning is a prominent branch in the field of artificial intelligence. It enables machines to analyze provided data and learn as they receive additional data over time. In this way, machines can automatically learn and improve gradually by utilizing data, without the need for being programmed explicitly. It has found remarkable possibilities across a wide range of sectors like, image processing, medical imaging, bioinformatics, autonomous vehicles, business decisions, market analysis, etc. A more sophisticated type of machine learning is deep learning. Deep learning models, from a perspective, are built systems influenced by biological brain. [2]. More formally, Deep learning enables computational models made up of multiple layers of processing to learn data representations

with varying degrees of abstraction. [3]. In the area of image processing, CNN, or Convolutional Neural Network, is one of the most sophisticated architectures in deep learning. CNN is a type of feedforward neural network that can extract features from input data using convolution architectures. [4]. It is made up of a sequence of convolutional layers linked to an array of hidden layers. The fundamental goal of the convolutional layers is to learn feature representations through the inputs. RCNN, which stands for Region based Convolutional Neural Network, can be seen as a combination of CNN and region proposals [5].

At present, image segmentation and object detection are two critical tasks in computer vision, with numerous potential applications including medical image analysis, scene understanding, image compression, etc. Segmentation is also a basic step in any medical image analysis for disease pattern recognition. A brain tumor, one of the most dangerous conditions, is an abnormal development of cells in the human brain. Brain tumor segmentation is a fundamental task that enables to separate the tumor area in a medical image for further study. For figuring out the precise location and size of the tumor, reliable brain tumor segmentation is necessary [6].

Over the years, several techniques for performing the segmentation task have been developed and established by different researchers. In image segmentation, certain architectures have recently achieved state-of-the-art performance. However, most of the architectures are complex in the sense that they involve an extensive number of training parameters. It makes them not only laborious to train and integrate but also increases the execution time of the task. Many of them allow using a pre-trained model in their workflow like Mask-RCNN [7]. But most of the popular pre-trained models have a huge number of training parameters, which also adds more to the complexity. Models such as AlexNet [8], ResNet [9], VGG16 [10], and others, for example, have a high number of parameters. These aspects make it challenging to fit them into compact devices with limited resources.

In our work, we aim to reduce the complexity of the segmentation process, while still maintaining considerable performance and accuracy compared to other existing architectures. The main idea here is that we make use of a low-complex RCNN architecture from [11] to classify the regions that contain the tumor area. The most appropriate region that covers the tumor area is selected based on the probability output and fed into the next stage. The next stage incorporates a lightweight implementation of U-Net architecture ideal for the segmentation task which is a modified version of the original U-Net [12]. This modification is aimed to lower the total number of parameters in our complete segmentation architecture. In the proposed approach, the system becomes less complex and more efficient with minimal loss of accuracy.

2 Literature Review

Several works on image segmentation were reviewed from a number of papers. RCNN (Region based Convolutional Neural Network) was introduced in [5],

which proposed a better approach to object detection. It achieved high object detection performance for its time. However, it had some limitations, such as an expensive multistage training pipeline, slow object detection, and so on. The author of [5] further improved the architecture in his subsequent work [13] named as Fast R-CNN, which eliminated some of the limitations of the previous work. In [14], the author upgraded the system even further by swapping the prior region proposal system with a new region proposal network that can be trained efficiently with the detection network. It was also limited to object detection task without segmentation, but it performed impressively. Several segmentation methods have been suggested over the years, such as Thresholding [15], Edge-based method [16], Region growing method [17], Watershed method [18], [19] and so on. Following that, substantial progress was made in image segmentation in [7] which can also generate object masks of detected objects in addition to the object detection task. It is a robust framework that allows any of the pre-trained models to serve as the backbone. U-Net, appeared in [12] is another remarkable architecture for performing accurate and rich medical image segmentation. U-Net achieves great performance and efficiency in the segmentation of medical images as shown in [12], [20] and also its variants in [21], [22], [23]. All of these architectures share one problem. Due to the large amount of trainable parameters available, they all require a very high system specification. For example, U-Net has nearly 34 million parameters for a similar configuration in our experiment. Researchers have proposed many techniques to reduce the count of parameters involved in segmentation without significantly reducing accuracy, such as in [24], [25] and so on. A recent work involving the brain tumor detection task with a low-complex architecture based on RCNN was successfully presented in the paper [11]. According to [11], it decreases the execution time for object detection and employs a low-complex model called Two-Channel CNN from the author's previous work, presented in [26], where the model was utilized to perform classification between two distinct types of brain tumor, called Meningioma and Pituitary tumors. In comparison to other existing architectures, the model has about 1,00,82,692 parameters [11] which is relatively reduced. However, this low-complex approach does not perform any segmentation task on the tumor MRI images. It only detects the tumor regions and outputs a bounding box or rectangle around the tumor-affected area. In short, it performs localization of the tumor in the MRI image with no segmentation. Our proposed architecture contributes to this work of [11] by extending it to the pixel-wise, precise segmentation of tumor area, while still preserving the key purpose: keeping the overall system less complex by reducing the total amount of trainable parameters inside. The proposed approach is basically divided into two sections: tumor region selection and segmentation. In the tumor region selection part, we use a nearly identical architecture as in [11]. Then we use a modified version of U-Net formed through the reduction of the number of blocks, filters and trainable parameters from the original version presented in the paper [12]. The optimal number of blocks and filters was determined by testing with different configurations in our

tests. Despite having a very small number of parameters, the proposed model has shown comparable accuracy and performance in our experiments.

3 Methodology

The two separate processes that make up the proposed architecture are region selection and mask generation. The Two-Channel CNN [11] serves as the core for the region selection, and a modified U-Net configuration is used for mask generation. This section describes these architectures and dataset processing.

3.1 Two-Channel CNN

The concept of using several channels and reducing hidden layers in order to minimize complexity has been evaluated in [27]. Similarly, to reduce complexity in our proposed architecture, we use the Two-Channel CNN from [11] where it serves as a feature extractor for the low-complex RCNN. Fig. 1 depicts the abstract diagram of the model from [11].

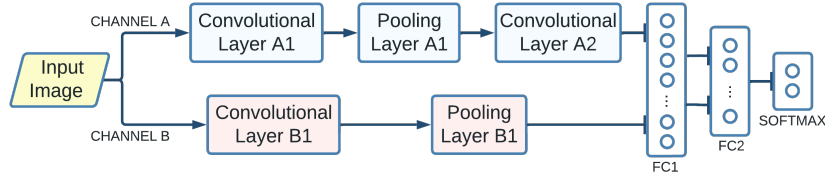


Fig. 1: Block diagram of the Two-Channel CNN [11]

According to [11], channel A has two convolutional layers denoted by A1 and A2. A max pooling layer is placed in between them. A1 has a 3×3 filter and the pooling layer size is 3×3 . The result is then passed on to A2 which also has a 3×3 filter. Channel B is slightly different than the earlier and it is made up of a convolutional layer B1 and a max pooling layer. In the work presented in [11], the filter size of B1 is 13×13 and dimension of the pooling layer is 3×3 . But for our proposed region selector, we fixed the filter size to 9×9 , since it showed better result in our experiments. These two channels are then concatenated and linked to two fully-connected layers, which are followed by a final softmax layer that is effective for classification task. The first layer that is fully connected has 100 neurons and the second one has 50 neurons which is similar to configurations stated in [26]. Previously in their work [26], Two-Channel CNN was used to classify Meningioma tumors and Pituitary tumors effectively. In our experiments, we tested it for the classification of tumor region and non-tumor region, which showed outstanding performance and accuracy in terms of total parameter count. Hence, we decided to use this Two-channel CNN in the region selection stage of our proposed model for the classification between tumor and non-tumor

regions. One major difference here is that we do not use any bounding box regressor in the proposed method unlike in [5], since it adds to the complexity and requires additional training tasks. For proper segmentation, the entire region needs to be contained inside the bounding box. Otherwise, a minor portion of the region might be cropped before the segmentation mask. To mitigate this issue, the bounding box was given a fixed 4-pixel padding to better fit the tumor region inside. While this affects the detection precision, the segmentation task is significantly improved without the additional cost of a bounding box regressor.

3.2 Modified U-Net

The original U-Net introduced in [12] had a contracting way on left side and also an expansive way on right side, each of which has 4 blocks. Each block in the contractive way employs double 3×3 convolutions, followed by ReLU and one max pooling of 2×2 . At each step of down-sampling, the count of feature channels is multiplied by 2. Each block in the expansive way starts with feature map up-sampling, then 2×2 convolution that reduces the number of channels in half, concatenating with feature map received from the contracting way, then double 3×3 convolutions and one ReLU. At the very last layer, a 1×1 convolution is used [12].

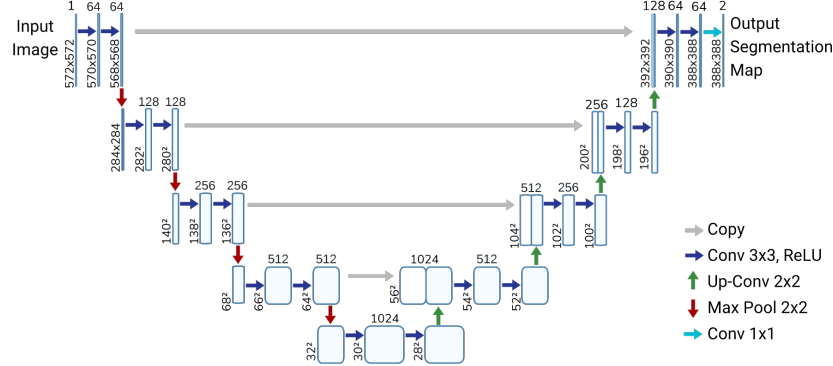


Fig. 2: U-Net architecture from [12]

We modified the U-Net architecture to have 3 blocks in the contractive and expansive paths instead of 4 blocks each, described in [12]. It is illustrated in Fig. 3. This configuration is selected based on the result of our experiments reflected in Table 4, which shows better overall results in terms of parameter count and performance. In this way, the number of filters is decreased and the count of parameters is also reduced by a significant amount. Based on these results, we are using this modified version for the segmentation task, in the mask generator phase of the proposed methodology.

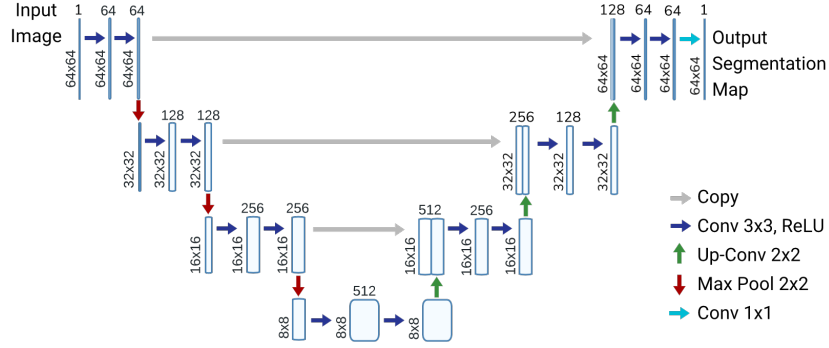


Fig. 3: Modified U-Net architecture in proposed method

3.3 Proposed Architecture

The proposed architecture is divided into two stages, region selector and mask generator, for the selected region. The block diagram of the proposed approach is shown in Fig. 4.

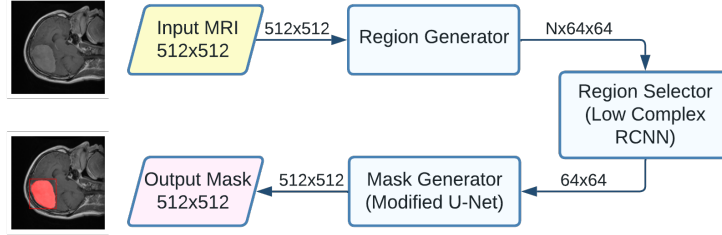


Fig. 4: Block diagram of our complete proposed architecture

Region Selector The workflow of the region selector is depicted in Fig. 5. Selective Search algorithm, described in [28] was used to produce all the regions. Then they are resized to size 64×64 and are fed to the Two-Channel CNN which classifies the regions containing tumor and healthy regions. Because of the applied Softmax activation in the final output layer, it gives the probability output. The most appropriate region to have tumor area is selected from this stage. Its bounding box co-ordinates are also recorded.

Mask Generator The selected tumor region of size 64×64 is then fed to the input of the modified U-Net. It gives a 64×64 sized output mask of the region, which is then resized back to its original size according to the previously-saved

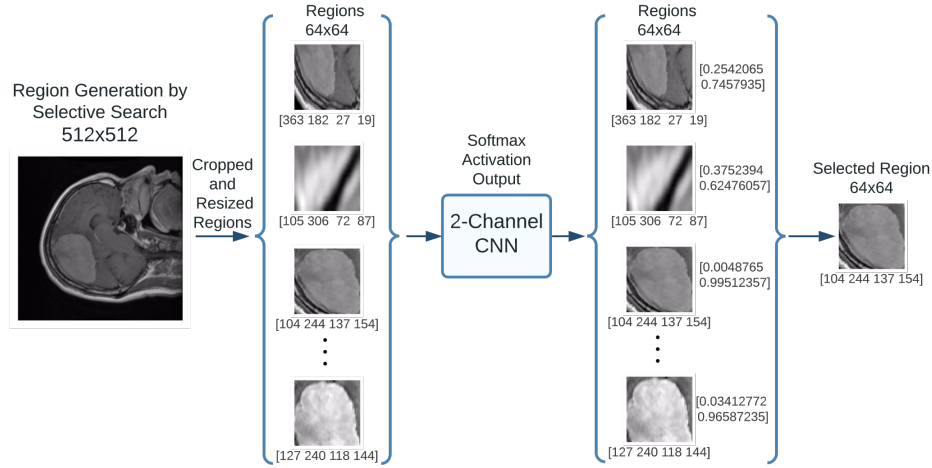


Fig. 5: Workflow of the region selector

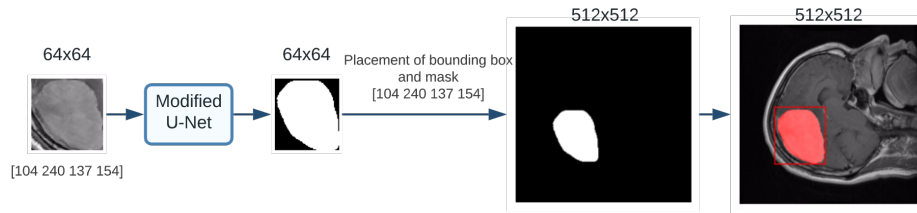


Fig. 6: Workflow of the mask generator

bounding box coordinates. After binary thresholding operation, the region mask is placed on the full 512×512 zero-valued image to form the complete final mask of the MRI image. Fig. 6 illustrates the workflow of the proposed mask generator.

3.4 Dataset Collection and Preprocessing

We used a dataset [1] which was also used in [11]. The dataset contains a total of 3064 MRI image slices (T1 contrast) from 233 patients. The dataset is divided into three classes: glioma, meningioma, and pituitary tumors. It contains 1426 MRI image samples from 92 individuals with Glioma, 708 MRI image samples from 82 victims with Meningioma, and 930 MRI image samples from 62 cases with Pituitary tumors. MRI slices are available in three orientations: axial, coronal, and sagittal. Fig. 7 depicts some examples. All of these images are grayscale and 512×512 pixels in size. We have only used the Meningioma tumor images from this dataset for our experiments since our main purpose is the segmentation task rather than the classification of tumor types. 5% of the Meningioma samples are set aside for testing. For the training part, we had to pick around 256 regions randomly from the generated regions for each of the MRI images

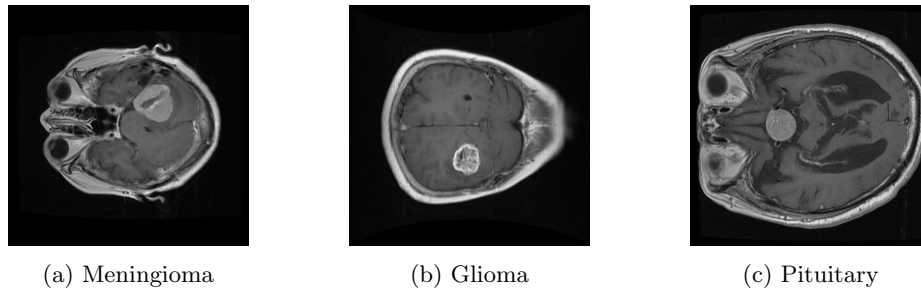


Fig. 7: Samples from the dataset [1]

using selective search algorithm. The ground-truth coordinates of the bounding box were calculated from the true mask of each MRI image. Hence, the regions are divided into two classes: one is tumor region covering the tumor area and another is non-tumor region. The class labeling of the regions is executed based on the IoU of the bounding box and rectangular region. During random region selection for each image, it is very important that the class distribution between the tumor-affected and non-tumor regions are balanced. Since the number of tumor-affected regions generated was lower than in the case of non-tumor regions, we had to consider regions with $IoU < 0.3$ as tumor-negative, $IoU \geq 0.7$ as tumor-positive regions. The regions having an IoU between 0.3 to 0.7 are considered positive if the ground-truth bounding box is inside the generated region. The conditions and considerations are illustrated in Table 1.

Table 1: Label considerations for generated regions based on IoU

Condition	Region label
$IoU < 0.3$	No tumor
$0.3 \leq IoU < 0.7$, ground truth region not inside	No tumor
$0.3 \leq IoU < 0.7$, ground truth region inside	Tumor region
$IoU \geq 0.7$	Tumor region

4 Result and Discussion

4.1 Experimental Setup

All of the tests are executed on Google Colab and Kaggle. All the results are based on the outputs from Kaggle using GPU P300 mode. On Local machine, Python, VS Code and Jupyter Notebooks are used for primary code generation and execution.

4.2 Classification of Tumor Regions

In our proposed region selector, Two-Channel CNN is used for the classification of tumor regions and healthy regions. From 673 Meningioma MRI images, a total of 172,288 regions were generated for the input. The regions were labeled using Table 1. They were preprocessed and resized to 64×64 before feeding into the region selector. 80% of the regions are used for actual training, and remaining 20% for validation of the classifier. The performance parameters listed below are used for evaluation:

$$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (1)$$

$$Specificity = \frac{True\ Negative}{True\ Negative + False\ Positive} \quad (2)$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (3)$$

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ P. + True\ N. + False\ P. + False\ N.} \quad (4)$$

$$F1\ Score = \frac{2 * Sensitivity * Precision}{Sensitivity + Precision} \quad (5)$$

The list of training parameters for region classification is provided in Table 2. The model was trained for up to 100 epochs, when we stop iteration since its validation accuracy and validation loss reached a satisfactory value. In our test, the initial accuracy was low, but it increased gradually up to a validation accuracy of 98.63%, having a final loss value of 0.0443. The training and validation curves are displayed in Fig. 8.

Table 2: Training parameters for region classification

Training parameter	Value
Learning rate	0.000001
Mini batch size	16
Data augmentation	No
Learning algorithm	ADAM
Epochs	100

Table 3, which exhibits the classifier’s performance on the test images, shows that the region classification task functions well. The segmentation architecture’s overall performance depends mostly on the classification task, and the result is satisfactory.

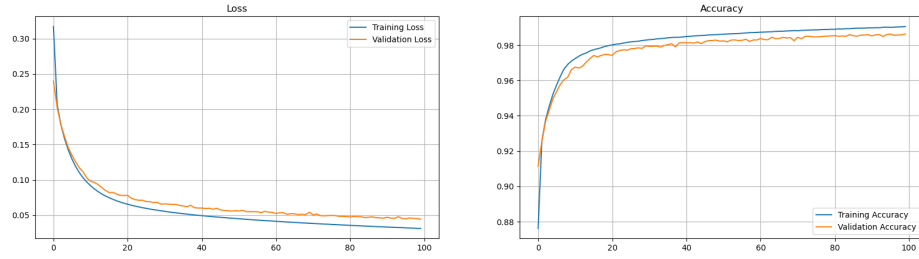


Fig. 8: Accuracy and loss curve for region classification

Table 3: Obtained result for region classification

Parameter	Obtained value
Sensitivity	99.08%
Specificity	98.19%
Precision	98.21%
Accuracy	98.64%
F1Score	98.64%
Total parameters	8,347,460

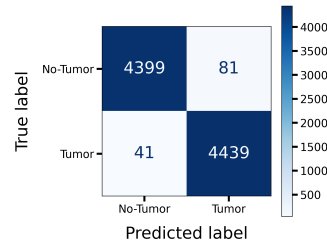


Fig. 9: Confusion matrix for region classification

4.3 Mask Generation from Selected Region

For mask generation, we used the modified U-Net architecture, described earlier. We modified the U-Net described in [12] by altering the number of blocks and the filters. In our experiment for the selection of optimal configuration, the training was performed for 10 epochs for each configuration. Besides, 10-fold cross validation method was employed for its evaluation. The obtained accuracy, IoU and dice coefficient are shown in Table 4. Despite the major reduction in parameters, the results are very much alike. This is due to the fact that the segmentation is carried out on regions that are much smaller than the entire image. Based on these metrics from Table 4, the optimal arrangement chosen is three blocks with no change in the number of filters. We have 8,556,353 parameters

with this configuration. When compared to the configuration of four blocks with half filters, the latter performs lower due to the reduced number of filters. Also, the remaining configurations are not optimal in terms of parameter count and performance. After selecting the optimal configuration, the model training was performed.

Table 4: Selection of optimal configuration for modified U-Net

Blocks	Filter reduced	Parameters	Accuracy	IoU	Dice score
3	By half	2,140,065	96.74%	92.61%	96.15%
3	No	8,556,353	97.73%	94.86%	97.35%
4	By half	8,629,921	97.69%	94.68%	97.25%
4	No	34,512,193	98.29%	96.16%	98.03%

In the training phase of the mask generation process, all 673 MRI images are used. Data augmentation is applied to generate a total of 10,768 region masks, from which 80% is utilized for actual training and the other 20% is used for validation. The following performance parameters are used for evaluating the modified U-Net:

$$IoU = \frac{\text{Area of overlap}}{\text{Area of union}} \quad (6)$$

$$\text{Dice coefficient} = \frac{2 * \text{Area of overlap}}{\text{Total area combined}} \quad (7)$$

Table 5: Training parameters for mask generation

Training parameter	Value
Learning rate	0.00001
Mini batch size	8
Data augmentation	Rotation & sclaing
Learning algorithm	ADAM
Epochs	100

The training parameters for mask generation are depicted in Table 5. From Fig. 10, it is clear that the training accuracy is low at the beginning, but it gradually increases with each iteration. The model went through training for a total of 100

epochs. The final validation accuracy is 97.11% and validation loss is 0.0708. Following training, the model was evaluated using test MRI images, and the results obtained are shown in Table 6. The updated model performed effectively on the segmentation task despite having a smaller number of trainable parameters. These performance values were calculated here considering the small region masks only. Later, for comparison, performance values were calculated from the complete image masks generated by the output of our proposed architecture.

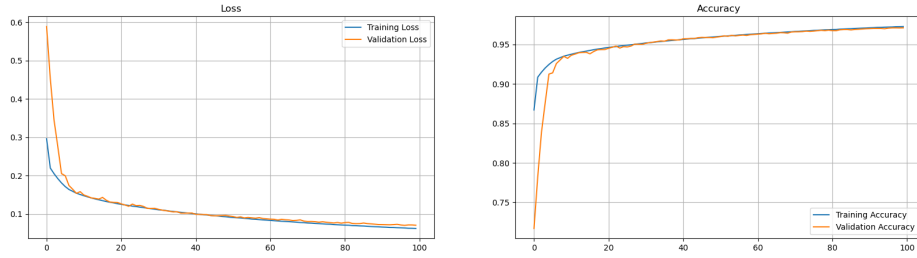


Fig. 10: Accuracy and loss curve for mask generator

Table 6: Obtained result for mask generation

Parameter	Obtained value
Accuracy	95.17%
Mean IoU	93.01%
Mean Dice Score	96.37%
Total parameters	8,556,353

4.4 Performance Evaluation of the Complete Architecture

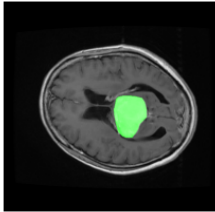
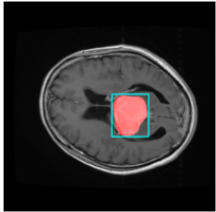
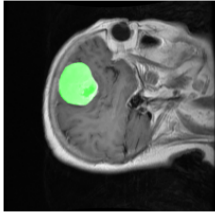
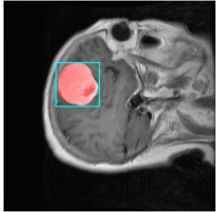
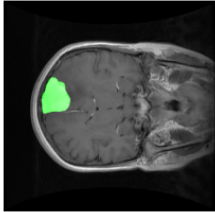
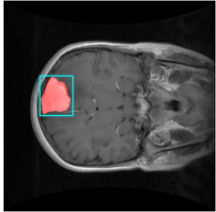
The complete architecture was tested against U-Net [12]. The original version performed perfectly, reaching 99.88% validation accuracy when we stopped training. Using the 35 randomly selected MRI images for testing, we evaluated our architecture with this model which produced 35 output masks. The results acquired from the test are provided in Table 7. It demonstrates that the accuracy of our proposed method is comparable to that of the original U-Net model. In terms of IoU and Dice Score, our proposed method outperformed it. According to the results, the number of total parameters in our proposed method is approximately half of those in the original U-net for this problem. However, it

Table 7: Result comparison between original U-Net and proposed architecture

Parameter	U-Net	Proposed method
Accuracy	99.79%	99.78%
Mean IoU	87.48%	89.76%
Mean Dice Score	93.11%	94.53%
Total parameters	34,512,193	16,903,813

performed well despite this decrease in the number of parameters. Because only the selected region is segmented, whose area is significantly smaller than the entire image. Therefore, our proposed approach is more efficient and less complex. Table 8 shows three photographs randomly picked from the 35 test images, along with their parameters and outputs generated by our proposed method.

Table 8: Segmentation result from proposed method

MRI sample	IoU	Dice score	True mask	Generated mask
Sample 1	94.00%	96.91%		
Sample 2	96.08%	98.00%		
Sample 3	94.46%	97.15%		

5 Conclusion

In this study, we presented a technique for efficient segmentation of Brain Tumor MRI images that generates a mask to separate the tumor area from the background. At first, our proposed method predicts and chooses the optimal tumor-containing region. The modified U-Net is then used to accomplish efficient and fast segmentation on the selected region. It is apparent that despite the reduced parameters, a comparable performance can be achieved using this region-only segmentation approach.

In terms of performance metrics, the system is likewise comparable to existing architectures. However, it has one drawback: the entire design requires two separate training processes for the region selector and mask generator, which is expensive. Furthermore, the regions are generated using a selective search algorithm, which takes a bit of time. With this minor tradeoff, the proposed approach performs admirably, as illustrated in 8. The results can be optimized further by using different lightweight segmentation models in the mask generator part, which has been kept for future improvement of this work.

In terms of complexity, the entire architecture is minimal. It is low-complex because it has a total of 16,903,813 parameters which is around half of the total parameters seen in the original U-Net design for this specific problem. But it shows comparable performance. Therefore, the proposed approach meets our primary goal of reducing the number of parameters while maintaining comparable performance.

References

1. J. Cheng, (2017). DOI 10.6084/m9.figshare.1512427.v5. URL https://figshare.com/articles/dataset/brain_tumor_dataset/1512427. (Accessed January, 2023)
2. I. Goodfellow, Y. Bengio, A. Courville, *Deep learning* (MIT press, 2016)
3. Y. LeCun, Y. Bengio, G. Hinton, *nature* **521**(7553), 436 (2015)
4. Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou, *IEEE transactions on neural networks and learning systems* (2021)
5. R. Girshick, J. Donahue, T. Darrell, J. Malik, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014), pp. 580–587
6. M. Ghaffari, A. Sowmya, R. Oliver, *IEEE reviews in biomedical engineering* **13**, 156 (2019)
7. K. He, G. Gkioxari, P. Dollár, R. Girshick, in *Proceedings of the IEEE international conference on computer vision* (2017), pp. 2961–2969
8. A. Krizhevsky, I. Sutskever, G.E. Hinton, in *Advances in Neural Information Processing Systems*, vol. 25, ed. by F. Pereira, C. Burges, L. Bottou, K. Weinberger (Curran Associates, Inc., 2012), vol. 25. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
9. K. He, X. Zhang, S. Ren, J. Sun, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778
10. K. Simonyan, A. Zisserman, arXiv preprint arXiv:1409.1556 (2014)
11. N. Kesav, M. Jibukumar, *Journal of King Saud University-Computer and Information Sciences* **34**(8), 6229 (2022)

12. O. Ronneberger, P. Fischer, T. Brox, in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III* 18 (Springer, 2015), pp. 234–241
13. R. Girshick, in *Proceedings of the IEEE international conference on computer vision* (2015), pp. 1440–1448
14. S. Ren, K. He, R. Girshick, J. Sun, *Advances in neural information processing systems* **28** (2015)
15. H. Zaidi, I. El Naqa, *European journal of nuclear medicine and molecular imaging* **37**, 2165 (2010)
16. A. Aslam, E. Khan, M.S. Beg, *Procedia Computer Science* **58**, 430 (2015)
17. G.C. Lin, W.J. Wang, C.C. Kang, C.M. Wang, *Magnetic resonance imaging* **30**(2), 230 (2012)
18. S. Pandav, *Int J Eng Res Technol* pp. 2278–0181 (2014)
19. N. Beheshti, L. Johnsson, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (2020), pp. 364–365
20. M.H. Hesamian, W. Jia, X. He, P. Kennedy, *Journal of digital imaging* **32**, 582 (2019)
21. A.Z. Atiyah, K.H. Ali, *IJEEEE Journal* pp. 21–27 (2021)
22. T. Tarasiewicz, M. Kawulok, J. Nalepa, in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 6th International Workshop, BrainLes 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Revised Selected Papers, Part II* 6 (Springer, 2021), pp. 3–14
23. K. Kotowski, J. Nalepa, W. Dudzik, in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 5th International Workshop, BrainLes 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 17, 2019, Revised Selected Papers, Part II* 5 (Springer, 2020), pp. 179–190
24. Y. Wang, Q. Zhou, J. Liu, J. Xiong, G. Gao, X. Wu, L.J. Latecki, in *2019 IEEE International Conference on Image Processing (ICIP)* (2019), pp. 1860–1864. DOI 10.1109/ICIP.2019.8803154
25. T. Emara, H.E.A.E. Munim, H.M. Abbas, in *2019 Digital Image Computing: Techniques and Applications (DICTA)* (2019), pp. 1–7. DOI 10.1109/DICTA47822.2019.8945975
26. N. Kesav, M. Jibukumar, in *Machine Learning for Predictive Analysis: Proceedings of ICTIS 2020* (Springer, 2021), pp. 119–131
27. N. Kesav, M. Jibukumar, *International Journal of Computers and Applications* **44**(10), 981 (2022). DOI 10.1080/1206212X.2022.2047443. URL <https://doi.org/10.1080/1206212X.2022.2047443>
28. J.R. Uijlings, K.E. Van De Sande, T. Gevers, A.W. Smeulders, *International journal of computer vision* **104**, 154 (2013)