3D U-Net Based Brain Tumor Semantic Segmentation Using A Modified Data Generator

Wagaye Tadele Kussa[1a], Yeabsira Mengistu Dana[1b], Hitesh Kag[1c], Dinesh Kumar[1d]

[a] ‘hagiosym@gmail.com’, [b] ‘yeabsiramengistu@gmail.com’, [c] ‘hitesh.kag@marwadieducation.edu.in’, [d] ‘dinesh.kumar@marwadieducation.edu.in’
[1] Department of Computer Engineering – Artificial Intelligence, Marwadi University, India

Abstract- Brain tumors, including gliomas, are a significant global health concern that causes numerous fatalities each year. Glioblastoma is a highly aggressive form of glioma that can have devastating symptoms. For optimal detection and therapy, brain tumor sections using multimodal magnetic resonance data must be segmented accurately. This study proposes a 3D U-Net semantic segmentation model with a Modified Data Generator (MDG) approach that leverages on-the-fly data augmentation to enhance model robustness. Our MDG approach generates unique data during training, which mitigates overfitting and improves generalization performance. Our model was trained and evaluated on the BraTS 2020 dataset, using the Dice score as our primary evaluation metric. Our approach achieved Dice scores of 82.2%, 90.3%, and 77.8% for the Tumor Core(TC), Whole Tumor(WT), and Enhancing Tumor(ET) subtypes, respectively, on the validation dataset, with low variation from the training data. We used an end-to-end training mechanism and did not employ transfer learning, which demonstrates our approach's effectiveness in segmenting brain tumor subtypes accurately and robustly. Our results suggest that the MDG approach is a promising method for improving the segmentation of brain tumor subtypes, which could improve patient outcomes and enhance brain tumor treatment.

Index Terms- 3D U-Net, Data augmentation, Deep learning, Segmentation of brain tumor, Semantic Segmentation, BraTS

1. Introduction

Results from the 2018 Cancer Registry[36] show that there are 18,078,957 instances of cancer in both sexes (Male and Female) have been recorded, of which 296,851 are connected to brain tumors. In 2018, there were 241,037 (2.71%) instances of mortality attributed to brain tumors out of 9,555,027 cases of cancer-related death. One of the most fatal brain cancers is glioma, which may seriously harm the neurological system and put patients at risk[1]. Due to the high frequency of glioma, early diagnosis, treatment, and intervention are popular study subjects[2]. Several MRI sequences are obtained in order to diagnose a brain tumor. T2-weighted with fluid-attenuated inversion recovery (FLAIR), T1-weighted (T1), T1-weighted with contrast enhancement (T1C), and T2-weighted (T2)[3].An example of a multimodal MR scan with a corresponding segmented glioma is shown below in **Figure 1**. The three glioma subregions of Enhancing tumor (ET), Edema (ED), and Necrotic core and Non-enhancing tumor (NCR/NET) display a variety of biological traits, along with three additional areas of interest: Tumor core (TC), Whole tumor (WT), and Enhancing tumor (ET)[4].

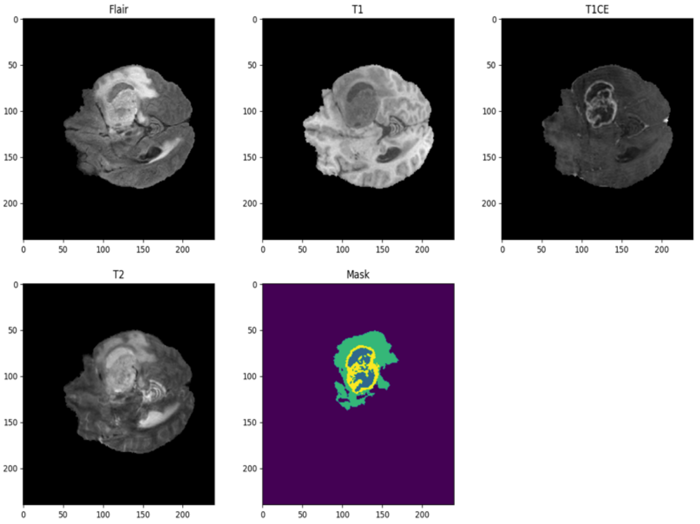


Figure 1: An example of multimodal MR images (Flair: First image, T1: Second image, T1CE: Third image, T2: Fourth image, and Mask: Last image); where the mask shows three partially overlapping interest areas (WT, TC, and ET) represented by TC: Blue, ET: Yellow,

Brain tumor segmentation manually from MR Scans is laborious and prone to human error. Researchers are working to create automated segmentation techniques that can precisely and effectively segment tumor parts from MR scans. Convolutional Neural Networks (CNN) is one such approach, where deep learning models have been demonstrated to have amazing performance in medical picture interpretation. Basic U-Net CNN architecture has been utilized and tweaked to create our 3D U-Net model. Olaf Ronneberger et at.[5] introduced the very first U-Net architecture in 2015. Due to its capacity to manage tiny, noisy structures and preserve spatial resolution during the segmentation process, the U-Net architecture is particularly well-suited for medical picture segmentation applications[6].

2. Dataset, Preprocessing, and Data augmentation

This project's MR pictures were from the BraTS 2020 challenge[7]. One folder contains the training dataset, while the other contains the validation dataset. There are 369 folders in total, with five nii files in each. Each of these nii files represents T1, T1CE, T2, FLAIR, and segmentation (Mask). Using the split folder library the training data was split into training and validation data. The validation folder provided by BraTS was only for online evaluation purposes, and the segmentation files were missing[8]. We performed the following preprocessing steps before we fed these data to our model.

We utilized the BraTS MR images that were originally 244x244 in size. However, to eliminate the non-informative regions surrounding the tumor, we performed patch extraction and cropped the images to 128x128. The resulting data consisted of 155 slices of 128x128 images per subject. To reduce the dimensionality of the data, we further discarded a few slices with no relevant information. To ensure the uniformity of the features, we applied a scaler.fit_transform() function to normalize the data and made the mean and variance zero and one respectively. The final shape of each modality was 128x128x128.

For the subsequent analysis, the data was loaded using the nibabel library and converted the data into arrays using the numpy library. The resulting arrays were saved as a single npy file. Finally, the data was split into 75% training and 25% validation data. The training dataset consisted of 258 npy images with the shape of (128,128,128,4) which included 4 modalities (T1, T1CE, T2, FLAIR) and 258 npy masks with the shape of (128,128,128,4) representing 4 tumor classes. The validation dataset consisted of 86 npy images with the shape (128,128,128,4), containing 4 modalities, and 86 npy mask files with the shape (128,128,128,4). The preprocessed data consisted of 344 folders in total.

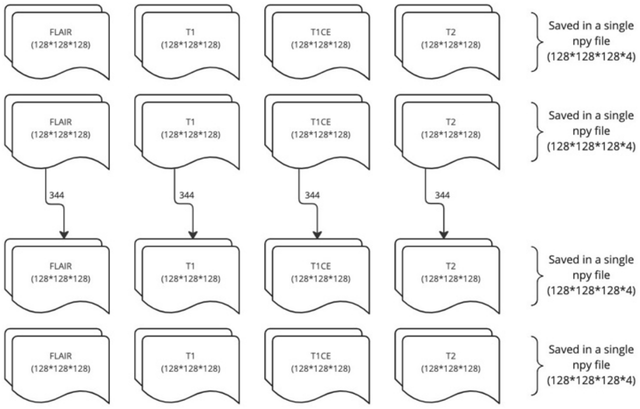


Figure 2: This is a representation of preprocessed training and validation images only (masks are not included here). Each line corresponds to a training image folder containing a single npy file which includes the four modalities in it. The same representation works for masks too, but the modalities are replaced by classes (WT, ET, TC, and Background).

After several attempts to train our brain tumor segmentation model with and without data augmentation, we introduced a novel approach called the Modified Data Generator (MDG) to improve model robustness and mitigate overfitting. We integrated the Volumentations 3D library by [31] with our custom-made data generator to generate augmented data on-the-fly during training. Our MDG approach ensured that each augmentation was carried out in a unique way, which allowed our model to learn from a diverse range of data and avoid overfitting. Unlike other approaches [32] that use convolutions before training or consume high amounts of memory, our MDG approach utilized a 16GB NVIDIA T4 TENSOR CORE GPU on Google Colab to train our model in a total of 59 epochs, which lasted approximately 88 hours with train, stop, and resume mechanism. Our model achieved outstanding segmentation results, as demonstrated by the improved Dice of the different tumor subtypes. The success of our MDG approach highlights its potential to enhance brain tumor sub-region segmentation and improve the effectiveness of brain tumor treatment.

Table 1: Illustration of details implemented during data augmentation in one of the six models trained

	Data augmentation steps used
Step 1:	Rotate in 0, 90, 180 and 270 P=0.5
Step 2:	Flip in axis 0 P=1/3
Step 3:	Flip in axis 1 P=1/3
Step 4:	Flip in axis 2 P=1/3

3. Related works

Deep learning has recently made incredible strides in a variety of medical picture segmentation tasks, especially when there is a wealth of training data available. Numerous multi-modal MR scans and related annotations of the tumor parts has been made public as part of the BraTS challenge[10]. Tumor segmentation task can be done using either two dimensional slice approach or three dimensional patch approach. In 2D CNN techniques, tumor parts are predicted independently for all slices of a 3D volume that has been partitioned into several 2D slices[11]. For instance, Caver et al.[12] used three different 2D U-Nets to segment WT, TC, and ET individually slice-by-slice.

Another common neural network approach for segmenting images is 3D U Net, which has an encoder/decoder structure and multiple connected convolutional layers in the encoder module. This approach aims to gradually minimize the feature map's spatial dimension and acquire more detailed semantic features that have been trained to be very effective at classifying images at a pixel level. The decoder component carries out an upsampling process on the layers, with the goal of reinstating both the object representation and spatial information[13]. J. Long et al. [14] described an inherent conflict between collecting semantic and spatial information because of the challenge of finding a model that can predict locally while taking general form into account. Some works solve this issue by combining both feature maps and adding "skip connections" across layers from the encoder and decoder modules. Combining local and spatial information is the basic concept. The advantage of skip connections was first utilized by Drozdzal et al. [15]. The memory and processing performance requirements for 3D U-Net are quite high. Therefore, using the complete 3D volume as the input and output may not be practical. One solution to this problem is to construct label maps for the smaller 3D patches by extracting them from the network input[16]. Many publishers indicated the use of a group of multiple 3D U Net models working jointly performs better than a single trained 3D U Net model. According to [17], A. Myronenko proposed the incorporation of a variational autoencoder into an encoder/decoder model, In order to recreate the input picture while segmenting it. This would force the network's previous layers to extract more useful features. In order to show that a trained U-Net network may produce competitive results, Isensee et al.[18] introduced the adoption of a 3D U-Net with small but significant alterations, such as the introduction of normalization(instance) [19] and ReLU(leaky). In another work by Zhou et al. [20] ensemble of several models was presented to segment the distinct tumor sub-regions successively while taking into account multiscale context data.

In a recent study by Xue Feng et al.[21], a model to segment brain tumors was constructed utilizing a 3D U-Net with modifications to training and testing methods, network architecture, and parameters of models. In order to eliminate random mistakes and enhance performance, they used a group of different models, and as a result, they placed ninth in the 2018 BraTS competition. According to reports, the Dice scores(mean) for the whole tumor (WT), tumor core (TC), and enhancing tumor (ET) were 87.80%, 75.40%, and 79.90%, respectively.

The BraTS 2020 training dataset was used in another study by Theophraste Henry et al. [22] to train several U-net-like neural networks to automate and standardize brain tumor segmentation. Techniques like stochastic weight averaging and deep supervision were used in the training. Brain tumor segmentation maps were produced using two distinct groups of models from various training pipelines, and the dice scores of each ensemble for certain tumor sub-regions was utilized to determine how the two ensembles should be combined. The final test dataset generated a Dice score for the solution of 79%, 89%, and 84%, placing it in the top 10 teams in BraTS 2020 challenge. The study also looked at more intricate training plans and neural network designs but found that they did not significantly boost performance and required longer training periods.

Here we have presented the performance result of the 3D U-Net semantic segmentation task[23] by Theophraste Henry et al. in a tabular form.

Table 2: Performance on BraTS 2020 dataset (by Theophraste Henry et al.)

Metric (mean)	ET	WT	TC
Dice(%)	78.507	88.595	84.273
Sensitivity	81.308	91.690	85.934
Specificity	99.967	99.905	99.964
Hausdorff in mm(95%)	20.36071	6.66665	19.54915

A contribution to the BraTS dataset is described in our base publication by Sarah Rosas Gonzalez et al. [33], with the goal of creating a precise and reliable system to segment brain tumor regions. In order to extract the most information from various visual modalities and combine them in an improved U-Net CNN network, the article suggests a multi-input module. This strategy seeks to enhance the effectiveness of brain tumor segmentation and prevent information loss of the modalities. The research significantly improves the accuracy of the model by extracting locally and globally distributed information using multiscale blocks inside the network. For the BraTS 2019 validation dataset, the suggested technique obtained dice values of 71.2%, 89.1%, and 79.3% for ET, WT, and TC, respectively. The reported total dice score was 79.8%, indicating the viability of the suggested strategy. With the use of a multi-input module and multiscale blocks inside the U-Net model, this study offers useful insights into how to improve brain tumor segmentation.

4. Method

The 3D U-Net model utilized in this paper uses the well-known U-Net architecture[24] which is crucial for segmentation tasks, as its foundation. The 3D U-Net model is intended to process image segmentation to remove the background from the target items or regions of interest after receiving 3D medical pictures as input. A number of Convolutional Transpose and Max-Pooling layers are used in the model to extract features, which are then refined and up-sampled by a number of Convolutional Transpose layers and Up-sampling layers. Additionally, the model has Batch Normalization layers to enhance performance and Dropout layers to avoid overfitting. To create the final segmentation result, the model includes a convolutional layer with a softmax activation function. The softmax activation function layer $[S(o(x))]_i$ explained by Jun Liu et al.[25] forces the output to be a probability.

$$[S(o(x))]_i = \frac{e^{[(o_i(x))]}}{\sum_i^n e^{[(o_i(x))]}} \qquad (1)$$

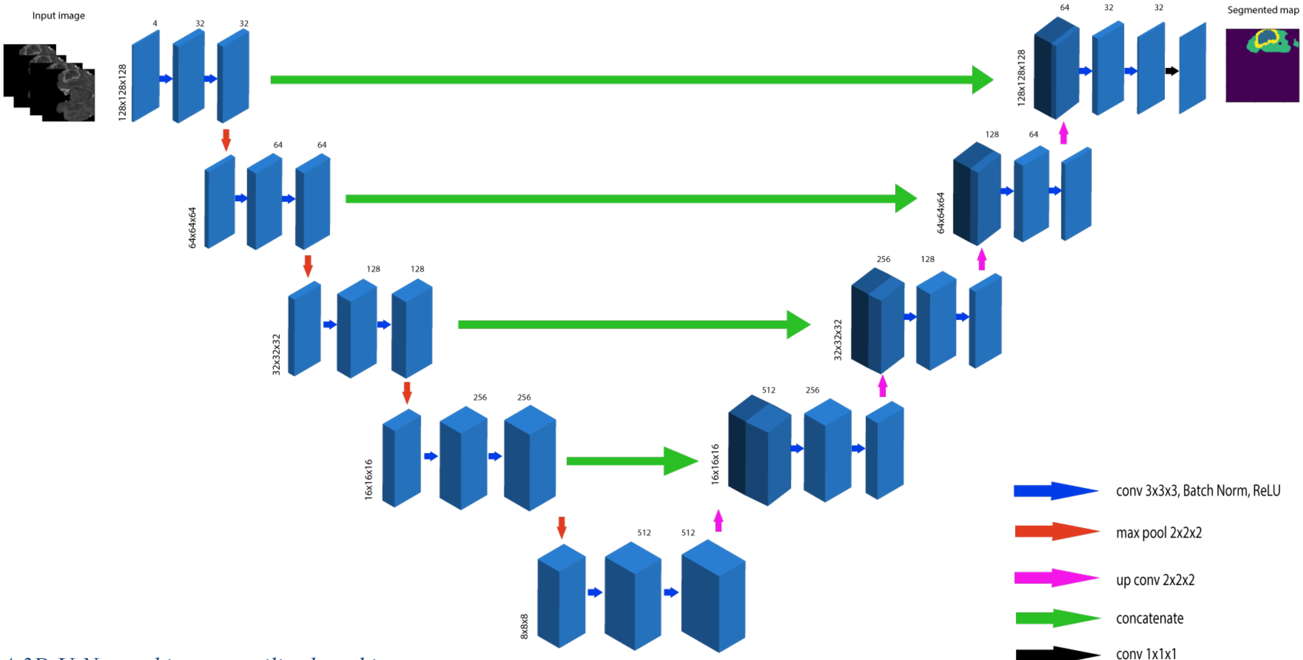


Figure 3: A 3D U-Net architecture utilized on this paper

4.1. Network Architecture

The 3D U-Net model architecture shown in **Figure 3** is composed of multiple blocks of convolutional, max pooling, and transpose convolutional layers, as well as batch normalization, activation, and dropout layers. The convolutional block contains two 3D convolutional layers, each with a (3,3,3) kernel size and a (1,1,1) stride. The number of filters used in all the convolutional layers can be specified using an argument. The outputs of the convolutional layers pass through a batch normalization layer if batch normalization is specified as True, and an activation layer with a ReLU activation function. The encoder network uses max-pooling layers to gradually reduce or down-sample the dimensions of the input. Then the max-pooling layers use a (2,2,2) pool size and a (2,2,2) stride. Dropout layers are also added when the max pooling layer is passed, to help prevent overfitting. The dropout rate can be specified using the dropout argument.

The decoder network utilizes the transpose operation on convolutional layers to increase the resolution of feature maps obtained from the encoder[13]. The transpose convolutional layers use a (3,3,3) kernel size and a (2,2,2) stride. Skip connections are employed after each transpose convolutional layer to concatenate the feature maps obtained from the decoder with their corresponding feature maps obtained from the encoder. This helps the network retain important information from the encoder while generating the final segmentation as explained by Drozdzal et al. [15]. The last layer of the network is a convolutional layer with a 1x1x1 kernel and a softmax activation function. This layer produces the segmentation mask that contains 4 distinct classes[25]. The 3D U-Net model takes a 3D input tensor and outputs a 3D tensor of segmentation masks with 4 classes. The architecture is flexible and can be adjusted based on the specific application requirements by modifying the number of filters, dropout rate, batch normalization, and other hyperparameters.

4.2. Evaluation metrics

The predictions are processed through a layer with soft-max activation that returns a probability value where each voxel belongs to the tumor region or the background. It is normal for the anatomy of interest to occupy only a relatively tiny portion of the scan in medical volumes like the ones we are analyzing in our work. As a result, the learning process frequently becomes stuck in the local minima of the loss function, producing a network with predictions that are heavily skewed in favor of the background[26]. In order to measure the similarity between the ground truth and predicted segmentation, we applied the Dice coefficient, a commonly used assessment metric in medical image segmentation tasks. In this work, the effectiveness of our 3D U-Net based brain tumor segmentation model was evaluated using the Dice Score. The Dice score is defined as the ratio of the number of voxels in the intersection of the predicted and ground truth masks, to the total number of voxels in both masks. Mathematically, the Dice coefficient per each voxel is represented as:

$$Dice\ Score = \frac{2\Sigma P_i G_i}{\Sigma P_i^2 + \Sigma G_i^2} \qquad (2)$$

Where the sums run over each voxel of prediction P_i and ground truth G_i[26]. In this implementation, the dice coefficient is calculated as the ratio of the sum of the product of the prediction and ground truth masks to the sum of the squares of both masks, with a small epsilon value added to the numerator and denominator to prevent division by zero. The mean Dice coefficient over all slices is returned as the final evaluation metric. In addition to the Dice score, we have also used the dice coefficient loss as an optimization objective to train our model. The dice coefficient loss is defined as one minus the Dice coefficient. Minimizing the dice coefficient loss during training would result in a model that produces masks with a higher degree of similarity to the ground truth masks. Mathematically, the Dice loss per each voxel is represented as:

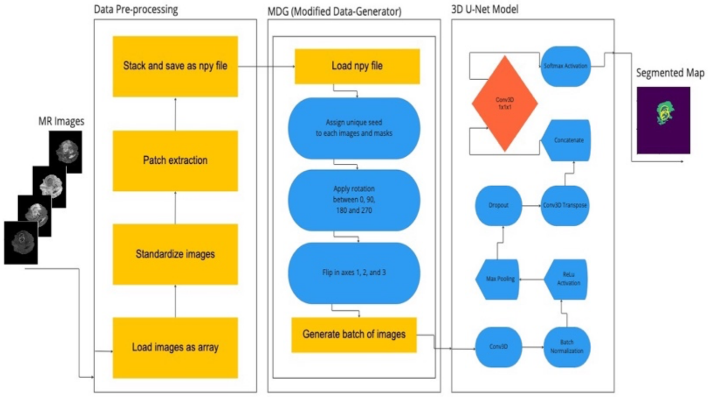
$$Dice\ Loss = 1 - \frac{2\Sigma P_i G_i}{\Sigma P_i^2 + \Sigma G_i^2} \qquad (3)$$

4.3. Data Augmentation and Data Generator

Due to ethics and data protection laws (such as GDPR), it is more challenging for medical imaging tasks than for other computer vision tasks to acquire a large number of training images. For this reason, data augmentation is even more crucial to increase the number of training images. There are surprisingly few studies that discuss the value of various augmentation techniques (such as rotations, random flips, scaling, and elastic deformations) for convolutional neural networks (CNNs) training, particularly for 3D CNNs, Marco Domenico Cirillo et al.[27]. The purpose of data augmentation is to expand the training dataset artificially by generating altered versions of the original images. By doing so, overfitting can be minimized, and the model's ability to generalize can be enhanced.

In the case of brain tumor semantic segmentation, data augmentation can help the model handle variations in imaging acquisition and variability in the appearance of the tumor. In this study, we implemented a modified data-generator (MDG) that involves several data augmentation techniques, including rotations at random angles of 0, 90, 180, or 270 with a probability of one and flipping at three axes with a one-third probability each, to the 3D MR images of the brain tumors used for training. This was done using an open-source custom library, Volumentations by Solovyev [28] and Albumentations by Buslaev et al.[29]. The rotation angle and flipping axes were chosen to increase the diversity of the augmented images and avoid over-augmentation.

To efficiently handle the large size of the 3D MR images, a data generator was implemented. A custom function was designed to load the images in batches and augment them on-the-fly during training and a random seed was set for each image to ensure that the augmentation was unique to each image. This resulted in a robust and accurate model with high performance on the validation data which is unseen during training. It also allowed efficient memory usage and faster training times, as the augmented images are not stored in memory. The data augmentation and data generator methods used in this study play an important role in improving the performance of the 3D U-Net model for brain tumor semantic segmentation.



Data preprocessing, step 2: Data augmentation step 3: Data generator, step 4: 3D U-Net model)

By utilizing these methods, it becomes possible to use resources effectively and reduce training times while simultaneously improving the model's generalization capacity[30].

5. Training and Results

For the training purpose, 75% of training and 25% of validation data were used. Our model was trained using a 16GB NVIDIA T4 TENSOR CORE GPU on Google Colab, and the training process was completed over 59 epochs, taking around 88 hours. Due to the session limit of Google Colab, a train-stop-and-resume mechanism was implemented. To assess the effectiveness of our 3D model, a function that calculates the Mean Dice score of each tumor subtype was defined. In addition, the Medpy library by [34] was also utilized, to compute the 95% Hausdorff distance in mm, as well as sensitivity and specificity for each tumor subtype. Adam with a starting learning rate of 0.0001 was also used as an optimizer, which reduces the learning rate by 20% when the training curve does not improve after five epochs. The Mean Intersection over Union (MeanIoU) was used to assess the model's performance. Overall, our training process was robust, and utilized various evaluation metrics to ensure the accuracy and generalization of our model.

The result section of our study evaluated the performance of our proposed MDG approach on the BraTS 2020 dataset. We compared our model's mean dice and 95% Hausdorff distance(mm) on WT, TC, and ET to some of the previous works done by Sarahi Rosas Gonzalez et al. [35]. Our model outperformed the other approaches Specifically, The dice scores attained by our model are 77.8%, 82.2%, and 90.3%, for ET, TC, and WT, respectively. Moreover, our model obtained 95% Hausdorff distances of 6.1, 7.4, and 7.5 mm for WT, TC, and ET, respectively. We also presented the results of all metrics used in a box plot, including sensitivity and specificity, to provide a comprehensive evaluation of our model's performance. The results below demonstrate that our MDG approach can achieve robust and accurate brain tumor segmentation on the BraTS 2020 dataset.

Table 3: Mean Dice score comparison of our proposed MDG 3D U-Net model and other models from [35]

Models	**Dice (%)**			**95% Hausdorff(mm)**		
	ET	WT	TC	ET	WT	TC
U+Net+ResCon +MSb [35]	66.2	88.2	74.7	6.7	6.9	8.8
U+Net+ResCon [35]	69.3	88.8	78.0	6.3	7.4	8.7
MSI U-Net [35]	71.1	89.0	79.2	**5.7**	7.4	7.7
MDG U-Net (Ours)	**77.8**	**90.3**	**82.2**	7.5	**6.1**	**7.4**

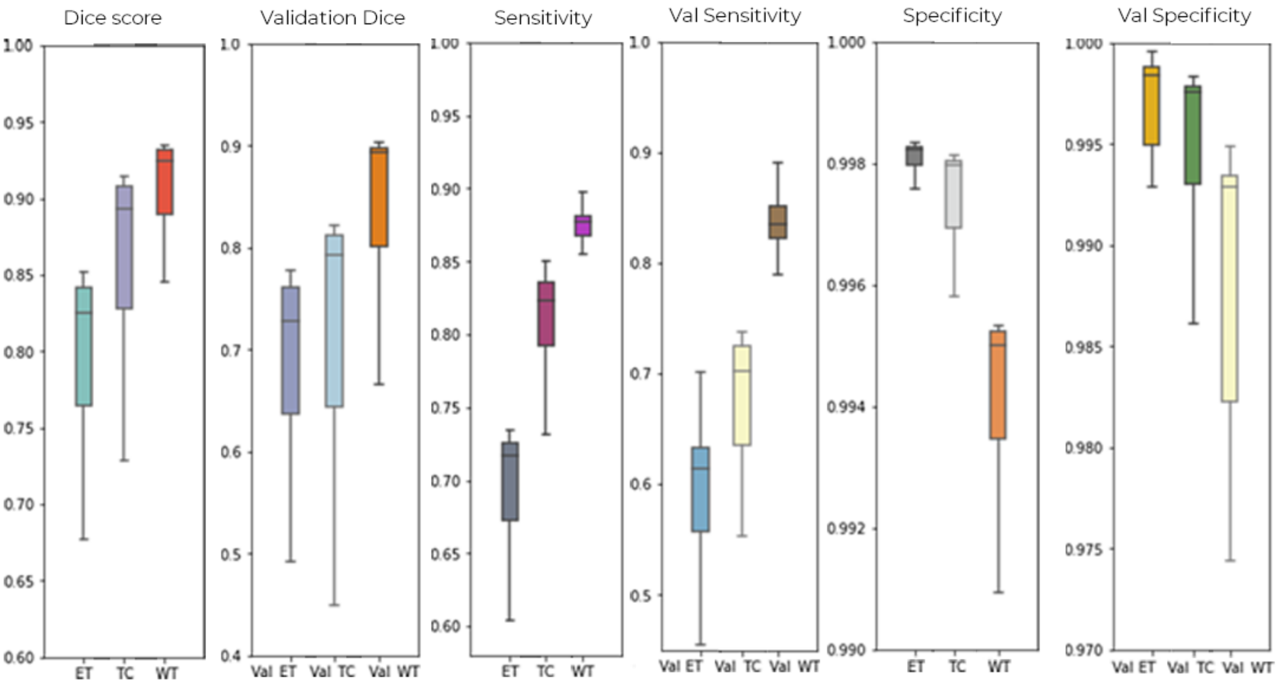


Figure 5: Box plot displaying Dice, Sensitivity and Specificity of training and validation data

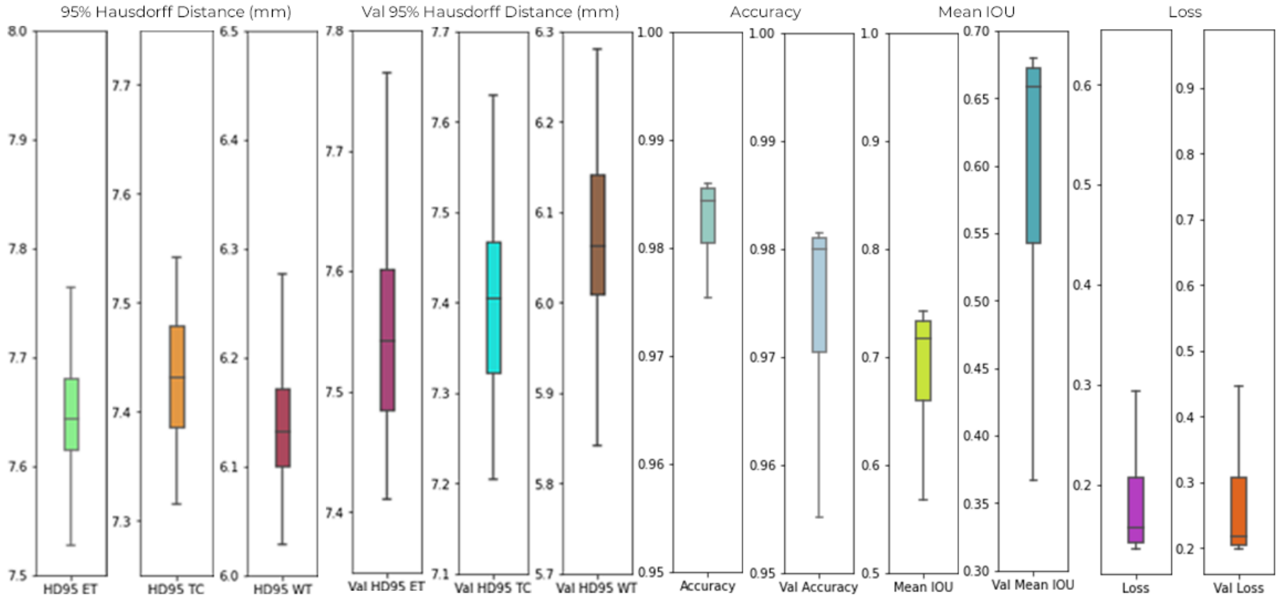


Figure 6: Box plot displaying the 95 percentile Hausdorff in mm, Accuracy, Mean IOU, Loss.

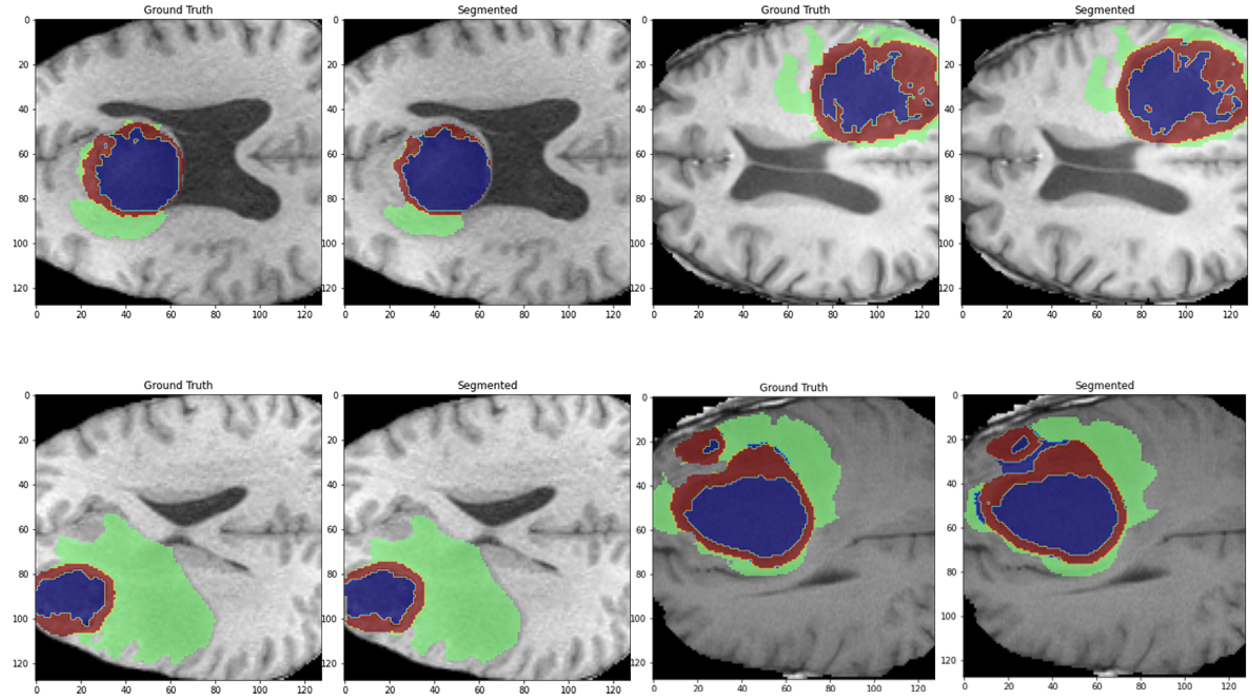


Figure 7: Segmentation results of our MDG 3D U-Net model Vs the ground truth; where, Necrotic & Non-enhancing tumor represented by Blue, Edema represented by Green and Enhancing tumor represented by Red.

6. Decision

In this work, we effectively used an MDG (Modified Data Generator) technique for 3D U Net-based semantic segmentation of brain tumors. Our findings showed that the use of MDG improved overall performance and addressed the problem of class imbalance, which has been a frequent problem in earlier research works. The precise segmentation of the enhancing-tumor portion was one of the biggest challenges we encountered. Nevertheless, by utilizing MDG, we were able to improve the segmentation results and narrow the difference between the training and validation curves. This shows that our model generalized effectively to new data and did not overfit to the training set of data. We may further enhance the performance of our MDG method by utilizing more advanced data augmentation techniques and including ensemble models in further research. Our research will likely be valuable in the creation of more precise and effective techniques for segmenting brain tumors. Overall, we expect that our research advances the discipline of medical image segmentation, especially with regard to brain tumors. In order to precisely detect and delineate tumor locations, which is crucial for therapy and diagnosis planning, we expect our technique becomes useful to the medical and research community.

7. Conclusion

Our study has shown that a 3D U-Net CNN may be used to create an end-to-end deep learning model for brain tumor segmentation. Despite the difficulties in segmenting tumors, our method has shown outstanding results without depending on weight transfer that has been previously learned. For the Whole Tumor, Tumor Core, and Enhancing Tumor subtypes, our approach resulted in mean dice scores of 90.3%, 82.2%, and 77.8%, respectively.

Acknowledgments

Support for this research paper was provided by the Marwadi University Faculty of Computer Engineering – Artificial Intelligence.

Authors

First Author – Wagaye Tadele Kussa, 4th-year Artificial Intelligence Student, Marwadi University, hagiosym@gmail.com.
Second Author – Yeabsira Mengistu Dana, 4th-year Artificial Intelligence Student, Marwadi University, yeabsiramengistu@gmail.com.
Third Author – Hitesh Kag, Assistant professor, Marwadi University, hitesh.kag@marwadieducation.edu.in.

Fourth Author – Dinesh Kumar, Associate professor, Marwadi University, dinesh.kumar@marwadieducation.edu.in.

References