Strategy Learning of Scaling Vision-Model 3D Volumetric Data in Biomedical Segmentation Task Brain Tumor: An Ensemble Learning Approach to BraTS 2020 Challenge

Haytham Al Ewaidat [1,*], **Youness El Brag** [2] **Ahmad Wajeeh Yousef E'layan** [3] **Ali Almakhadmeh**[4]

[1]Jordan University of Science and Technology, Faculty of Applied Medical Sciences, Department of Allied Medical Sciences-Radiologic Technology, Irbid, Jordan, 22110

[2]Abdelmalek Essaâdi University of Science and Technology, Faculty of Multi-Disciplinary Larache, Department of Computer Sciences, ksar el kebir , Morocco, 92150

[3,4]Jordan University of Science and Technology, Faculty of Applied Medical Sciences, Department of Allied Medical Sciences-Radiologic Technology, Irbid, Jordan, 22110

Correspondence author: Dr Haytham Al Ewaidat, Department of Allied Medical Sciences-Radiologic Technology, Faculty of Applied Medical Sciences, Jordan University of Science and Technology. PO Box 3030, Irbid 22110, Jordan Tel: (+962)27201000-26939; Fax: (+962)27201087; E-mail: haewaidat@just.edu.jo

Conflict of interest: The authors declare no conflict of interest of any type.

Data availability: The data and code for this article is available on GitHub: The code implementation can be found in our GitHub repository: https://github.com/deep-matter/AttentionUnetPlus_EnsembleLearning.

Funding: This work is supported by the Jordan University of Science and Technology, Irbid-Jordan, under grant number 20200649

Abstract

Purpose: Accurate segmentation of brain tumors is critical for patient treatment and prognosis. The purpose of this study is to show different Staratgy learning to Train multiple models with different Hyper-Paramters selection and Loss functions which lead into enhance the performance model in the Last stage we used an ensemble learning approach for brain tumor segmentation using the BraTS to demonstrate different Staratgy learning can provide the significant advantage of Strategy learning 2020 dataset.
Approach: Two segmentation models, 3D U-Net++ and 3D U-Net++ with attention gate, are trained using different learning strategies on the BraTS 2020 dataset. Hyperparameters are adjusted, and diverse loss functions are employed. The models segment gliomas into the whole tumor (WT), tumor core (TC), and enhancing tumor (ET) regions. Performance is evaluated using dice similarity coefficient (DSC) and Jaccard similarity coefficient (JSC). Outputs from the individual models are combined using ensemble learning with weighted voting.
Results: Different learning strategies yield varied performance for the segmentation models. The ensemble learning approach with weighted voting improves performance compared to some individual models. On the BraTS 2020 validation set, the ensemble model achieves the following DSC and JSC values: WT DSC ±0.86, TC DSC ±0.86, ET DSC ±0.71, WT JSC ±0.77, TC JSC ±0.77, and ET JSC ±0.57. The ablation study demonstrates the importance of leveraging different learning strategies for the ensemble. and found that both models were important for achieving optimal performance.
Conclusion: In our research, we have shown that using different strategy learning is highly effective for accurately segmenting brain tumors. By combining multiple segmentation models. Ensemble learning has the capability to improve the performance of a certain model by achieving greater accuracy than what can be achieved by a single model on its own and be stable for previous performance This approach has significant potential to enhance clinical decision-making for individuals with brain tumors

Keywords: ensemble learning, U-Net++, attention gate, Convolutional Neural Networks, weighted voting scheme, brain tumor segmentation.

1 introduction

Gliomas are primary brain tumors that can be categorized as high-grade or low-grade based on their clinical presentation.[1,2] High-grade gliomas (HGG), such as glioblastoma multiform (GBM), are aggressive and invasive tumors that can lead to the patient's death in a short period of time. On the other hand, low-grade gliomas (LGG) are slow-growing tumors with a longer life expectancy.[3]

The standard treatment protocol for gliomas involves surgical removal of the tumor followed by radiation therapy. The goal of radiation therapy is to irradiate the tumor volume while minimizing damage to surrounding normal tissues.[4] This requires accurate determination of the 3D treatment volumes, which traditionally involves a manual procedure of outlining the gross tumor volume (GTV) on multiple 2D imaging slices using CT or MRI.[5]

However, this process is time-consuming and prone to variability and uncertainty due to the complex nature of gliomas. Therefore, there has been a recent focus on improving target volume definition methodology through the use of advanced imaging modalities. Despite these efforts, there is still a need for better techniques to accurately and efficiently segment gliomas.

Deep learning-based methods using convolutional neural networks (CNN) have shown significant

progress in multi-modal brain tumor segmentation. The 3D CNN architectures, including 3D U-Net and 3D U-Net++, have demonstrated superior performance in capturing 3D contextual information. 3D U-Net++ has additional dense skip connections for better information flow between the encoder and decoder. Attention mechanisms have also been integrated into U-NET model, such as the 3D attention U-Net, which uses attention gates to selectively emphasize informative features and suppress irrelevant features for better segmentation performance.

Another popular approach for improving segmentation performance is ensemble learning, which combines the outputs of multiple models to obtain a more accurate and robust segmentation. Ensemble learning can be done at the model level or the output level. At the model level, different architectures, initializations, or training data can be used to train multiple models, which are then combined by averaging their predictions or using a majority voting scheme. At the output level, the same architecture and training data are used to train multiple models with different random seeds or augmentation schemes, and their predictions are averaged or combined using more sophisticated methods, such as conditional random fields. Ensemble learning has been shown to improve the performance of deep learning-based segmentation methods for brain tumors and other medical applications.

In this context, addressing algorithmic uncertainty in tumor segmentation is crucial for improving treatment planning and patient outcomes. Recent research has explored the use of ensemble learning approaches for improving the accuracy of glioma segmentation.[6,7] These methods combine multiple models to reduce uncertainty and improve segmentation results.

2 Related Work

2.1 Early Methods for Brain Tumor Segmentation

In the past, numerous automatic methods have been proposed for the segmentation of brain tumors. Early reported methods mainly relied on the extraction of handcrafted features that represent different tissues or on registration to an anatomical template. Prastawa et al.[8] proposed a method that can segment brain tumors simultaneously with the detection of edema. Similarly, Gooya et al.[9] presented an approach for joint segmentation and deformable registration of brain scans of glioma patients to a normal atlas. Although these traditional segmentation methods have achieved acceptable performances, they still suffer from limited accuracy due to the complexity of the brain tumor heterogeneity

2.2 Deep-Learning-Based Methods for Brain Tumor Segmentation

Recently, with the advancement of deep learning, numerous deep neural network-based methods have been developed for brain tumor segmentation. These methods usually adopt an end-to-end structure and perform pixel-wise prediction. For instance, Kamnitsas et al.[10] proposed a method that ensembles different models and architectures for robust performance through the combination of predictions from various methods. Similarly, Wang et al.[11] developed a cascaded approach to decompose the multi-class segmentation problem into a sequence of three binary classification tasks. Despite the superior performance of these deep learning-based methods, the problem of algorithmic uncertainty in tumor segmentation still persists. Therefore, addressing this problem remains a crucial challenge in the field.

Adding to this, the current promising method for addressing the algorithmic uncertainty in brain tumor segmentation is the ensemble learning approach. In this approach, multiple models are trained with different initialization conditions and hyperparameters, and their predictions are combined to achieve robust performance. The ensemble learning approach has been shown to outperform the single-model approach in many segmentation tasks, including brain tumor segmentation. In this regard, the proposed method in this paper aims to address the algorithmic uncertainty in tumor segmentation by leveraging the power of ensemble learning. Specifically, we aim to investigate the effectiveness of ensemble learning in improving the segmentation performance of the BraTS 2020 dataset.

In this paper, we propose different strategies of learning to address algorithmic uncertainty in tumor segmentation using the BraTS 2020 dataset. We evaluate our method on the BraTS 2020 validation set and demonstrate its effectiveness. Our approach combines two different deep learning models with different architectures, loss functions, and data augmentation strategies. We then apply the weighted voting algorithm to fuse one single output of these models and obtain the final segmentation result. Our ensemble approach is designed to capture complementary information from multiple models and mitigate the effects of algorithmic uncertainty. We also perform an ablation study to demonstrate the effectiveness of using different strategies of learning and compare our ensemble approach within individual models

3 Material and Methods

3.1 Dataset

The BraTS-2020 dataset[12–14] consists of two sets, namely training, and validation, which are used for developing and testing brain tumor segmentation models.The BraTS multimodal scans are provided in the form of NIfTI files (.nii.gz) and include the following volumes: a) native (T1), b) post-contrast T1-weighted (T1Gd), c) T2-weighted (T2), and d) T2 Fluid Attenuated Inversion Recovery (T2-FLAIR). These scans were obtained using different clinical protocols and scanners from multiple institutions (n=19), which are acknowledged as data contributors.

Each imaging dataset has undergone manual segmentation by one to four raters following a consistent annotation protocol. The annotations were carefully reviewed and approved by experienced neuro-radiologists. The provided annotations encompass the GD-enhancing tumor (ET — label 4), peritumoral edema (ED — label 2), and necrotic and non-enhancing tumor core (NCR/NET — label 1. Overall, the BraTS-2020 dataset provides a diverse range of brain tumor cases with different grades and modalities, making it a valuable resource for developing and testing brain tumor segmentation models.

3.2 Post-Preprocessing and Data-Augmentation

In our data post-processing stage, we dealt with 3D volumetric data. we split the data using Stratified K-fold[15] to balance the data in which the number of folds is 7, and instead, we trained both of the models with a given fold zero. data augmentation we used Torch.io's[16] built-in methods to improve data representation, including :

1. RandomBiasField: is a technique used to correct for the effects of magnetic field inhomogeneity in MRI images. It works by modeling the magnetic field inhomogeneity as a

smooth, spatially varying bias field, and then estimating this bias field from the MRI data using mathematical algorithms. By correcting for the spatially varying bias field, the RandomBiasField[17] technique can reduce the intensity variations caused by magnetic field inhomogeneity in the MRI image, improving image quality and making it easier for medical professionals to interpret the image.

2. RescaleIntensity: this is a technique used to adjust the brightness or contrast of an MRI image by rescaling the intensity values of the pixels within the image. This technique is commonly used to improve the visual appearance of MRI images and make them easier to interpret

3. Normalization: is a technique used in image processing to improve contrast and enhance features within an image. The method works by scaling the intensity values of an image so that they fall within a specific range, typically between 0 and 1. The technique is useful in situations where the intensity values of an image are not evenly distributed or are skewed towards one end of the scale.

$$\text{Normalization } = \frac{x - x.min()}{x.max() - x.min()} \tag{1}$$

Where:

- x is the original intensity value of a pixel within an MRI image
- $x.min()$ is the minimum intensity value of all pixels within the MRI image
- $x.max()$ is the maximum intensity value of all pixels within the MRI image

4. Resized: we resized the whole samples of data into 50x50x50 to ensure the model was trainable on the environment we set up, which provided us with free access to multi-GPUs for training,

Finally, We then extracted the channels of the three classes we were interested for segmenting the Region of interest (ROI) which includes the whole tumor (WT), tumor core (TC), and enhancing tumor (ET). The post-processing stage aimed to improve the quality of the data and facilitate the segmentation process.

3.3 Models architecture

In our approach, we used two models for brain tumor segmentation: 3D U-Net++[18] and 3D Attention[19] U-Net++. These models were trained independently on the BraTS dataset to obtain the best-saved weights for use in the final stage of the ensemble learning process. We then used a weight voting algorithm to combine the predictions of these two models and improve the segmentation performance. This approach has been widely used in ensemble learning to achieve higher prediction accuracy by combining the outputs of multiple models.

3.3.1 U-Net-Plus-PLus Model

The architecture of U-Net++ comprises an encoder and decoder that are connected through a series of nested dense convolutional blocks. U-Net++ aims to bridge the semantic gap between the

feature maps of the encoder and decoder before fusion. The re-designed skip pathways are the main distinguishing factor between U-Net++ and U-Net[20] (black components in Figure 1a). These skip pathways help to transform the connectivity of the encoder and decoder sub-networks. The dense convolution blocks, with different numbers of convolution layers, bridge the semantic gap between the feature maps. U-Net++ also incorporates deep supervision, as shown in red in the graphical abstract Figure 1 . Interestingly, U-Net++ can be pruned at inference time if trained with deep supervision.

(1) Re-designed skip pathways: In U-Net++ , the skip pathways between the encoder and decoder undergo a dense convolution block that transforms their connectivity. This block has a variable number of convolution layers, depending on the level of the pyramid. Each layer is preceded by a concatenation layer that fuses the output from the previous convolution layer of the same dense block with the corresponding up-sampled output of the lower dense block. This brings the semantic level of the encoder feature maps closer to that of the decoder feature maps, which makes optimization easier. The output of each node along the skip pathway is denoted as $x^{i,j}$, where i is the down-sampling layer index and j is the convolution layer index. The stack of feature maps represented by $x^{i,j}$ is computed as :

$$x_{i,j} = \begin{cases} H(x_{i-1,j}) & j = 0 \\ H([[x^{i,k}]_{k=0}^{j-1}, U(x^{i+1,j-1})]) & j > 0 \end{cases} \tag{2}$$

The equation for this skip pathway uses the function $H()$ for convolution followed by an activation function, $U()$ for up-sampling, and [] for concatenation. Nodes at level $j = 0$ receive only one input, nodes at level $j = 1$ receive two inputs, and nodes at level $j > 1$ receive $j + 1$ inputs, with j inputs from the previous j nodes in the same skip pathway, and the last input is the up-sampled output from the lower skip pathway.

(2) Deep supervision: Our proposed approach in U-Net++ is to use deep supervision,[21] which allows the model to operate in two different modes. The first mode is called the accurate mode, where the outputs from all segmentation branches are averaged. The second mode is called the fast mode, where the final segmentation map is selected from only one of the segmentation branches. The choice of segmentation branch determines the extent of model pruning and speed gain. Figure 1c illustrates how the selection of a segmentation branch in fast mode leads to different architectures with varying complexity.

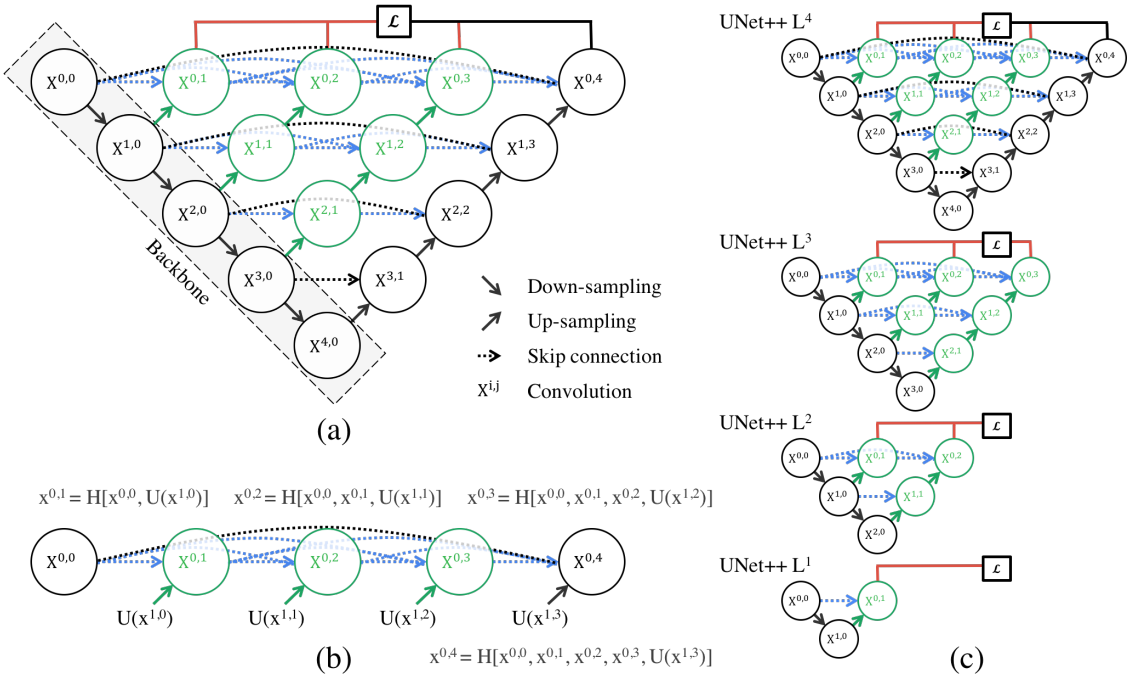


Fig 1: This Figure provides a detailed overview of the U-Net++ architecture, which includes an encoder and decoder that are connected by a series of nested dense convolutional blocks. The primary objective of U-Net++ is to address the semantic gap between the feature maps of the encoder and decoder before fusing them

3.3.2 Attention U-Net++ Model

Attention Nested U-Net++ is an extension of U-Net++ that includes attention gates in the skip connections of the network, resulting in better feature representation and improved segmentation performance. The attention gates are added to the dense convolutional blocks in each skip connection, allowing the model to learn which features from the previous level are most important for the current level. In addition to attention gates, Attention U-Net++ also includes a nested structure with multiple levels of encoders and decoders, each with their own set of dense convolutional blocks and attention gates. This nested structure further enhances the model's ability to capture more relevant features. Overall, the attention gates and nested structure in Attention U-Net++ improve the accuracy of brain tumor segmentation and have the potential to enhance the clinical decision-making process.

(1) The Attention Gate (AG)[22] is a mechanism that originated in natural language processing (NLP) but has recently been applied to computer vision. He Kaiming's team first introduced the attention mechanism to computer vision with their Non-local model.[23] Since then, researchers have combined shared networks with attention mechanisms for semantic segmentation and incorporated attention mechanisms with residual networks to obtain deeper networks .

To focus on relevant locations for the target organ,The Attention Gate takes the upsampling features and the corresponding down-sampling features as inputs, then uses a squeeze-and-excitation (SE) block to compute channel-wise attention maps. These attention maps are then multiplied element-wise with the original up-sampling features, providing the network with

more precise and relevant information for segmentation. The architecture of the Attention Gate is shown in Figure 2, and its effectiveness in improving segmentation performance has been demonstrated in various studies. By incorporating Attention Gates into U-Net++,

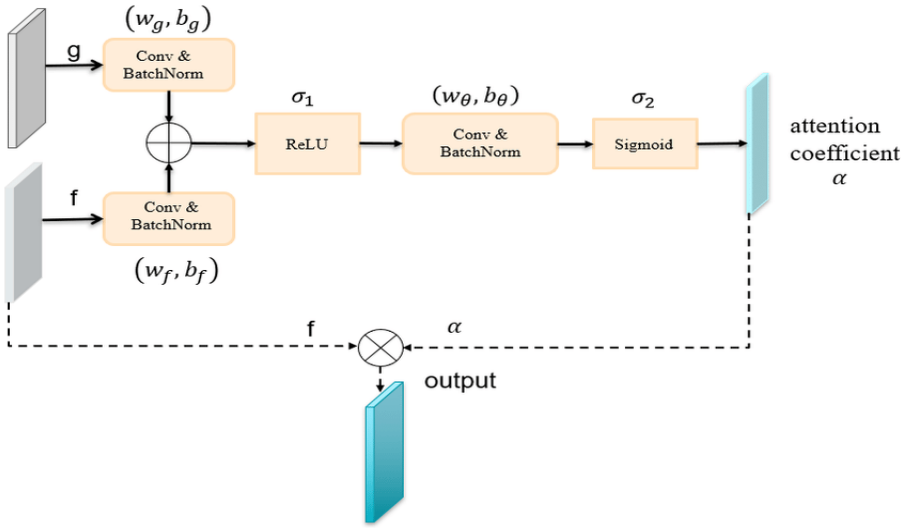


Fig 2: The Attention Gate is a simple yet effective mechanism that focuses on relevant locations in the image.

(2) Attention U-Net++ : is a medical image segmentation network that utilizes nested U-Net as its basic framework. It features symmetrically arranged encoder and decoder networks with dense skip connections that propagate context information to extract efficient hierarchical features. The network also includes attention gates in the skip connections to select important features. The extracted feature map of a convolution layer is defined by $\Phi[]$ with concatenation merger, $Up()$ for upsampling, and $Ag()$ for attention gate selection , so extracted feature map of the convolution layer can be defined as:

$$X_{i,j} = \begin{cases} \Phi[X_{i-1,j}] & \text{if } j = 0 \\ \Phi\left[\int_{k=0}^{j-1} Ag(X_{i,k}), Up(X_{i+1,j-1})\right] & \text{if } j > 0 \end{cases} \tag{3}$$

Figure 3 provides a detailed analysis of the first skip pathway in Attention U-Net++

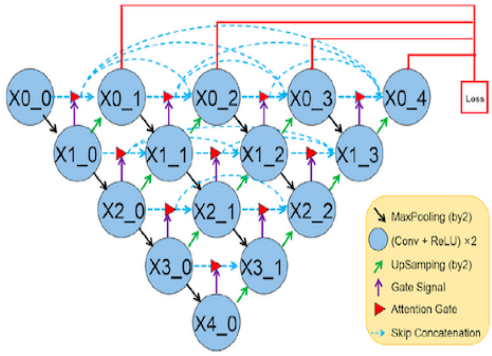


Fig 3: The Attention Gate is a simple yet effective mechanism that focuses on relevant locations in the image.

(3) Deep supervision: is incorporated in Attention U-Net++ to improve the model's performance. This is achieved by adding a 1x1 convolutional layer and a sigmoid activation function after every output node $X_{0,1}, X_{0,2}, X_{0,3}, X_{0,4}$ as shown in Figure 4. The dense skip connections in the nested blocks enable Attention U-Net++ to obtain full-resolution feature maps at different semantic levels from the nodes. To effectively integrate these semantic information, a hybrid loss function that combines soft dice coefficient loss.

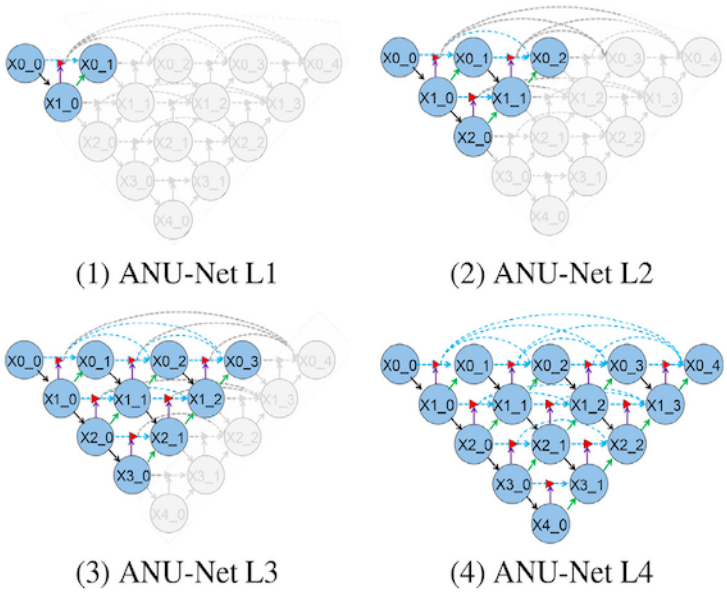


Fig 4: rained with deep supervision, Attention U-Net++ enables segmentation at multiple levels. Gray regions indicate removed nodes and attention gates during prediction.

3.3.3 Ensemble learning

The weighted voting ensemble learning algorithm has gained widespread popularity due to its simplicity, intuitiveness, and effectiveness. It combines various base learners and trains new learners with weights to balance the shortcomings of a group of high-performing models. However, the success of ensemble learning hinges heavily on the diversity of the base learner outputs and the methods used to consolidate these outputs into a single result.

In our study, To enhance tumor segmentation accuracy using the BraTS 2020 dataset, we employ a weighted voting ensemble learning algorithm. This algorithm has gained popularity for its simplicity, intuitiveness, and effectiveness. By combining multiple base learners and assigning weights to new learners, we can address the limitations of individual models. we merge two deep learning models with different architectures and loss functions. Through the weighted voting scheme, we obtain the final segmentation result. Ensemble learning has the potential to improve model performance by surpassing the accuracy achieved by a single model. Additionally, it provides stability in performance across various scenarios.

Our proposed ensemble learning approach demonstrates performance compared to individual models when evaluated on the BraTS 2020 validation set. Based on statistical analysis, we identify the best learning strategy to follow, ensuring the selection of an optimal approach.

By leveraging strategies of learning, we mitigate algorithmic uncertainty and achieve enhanced tumor segmentation results. This research contributes to the field by emphasizing the benefits of strategies of learning and statistical analysis in improving segmentation accuracy.

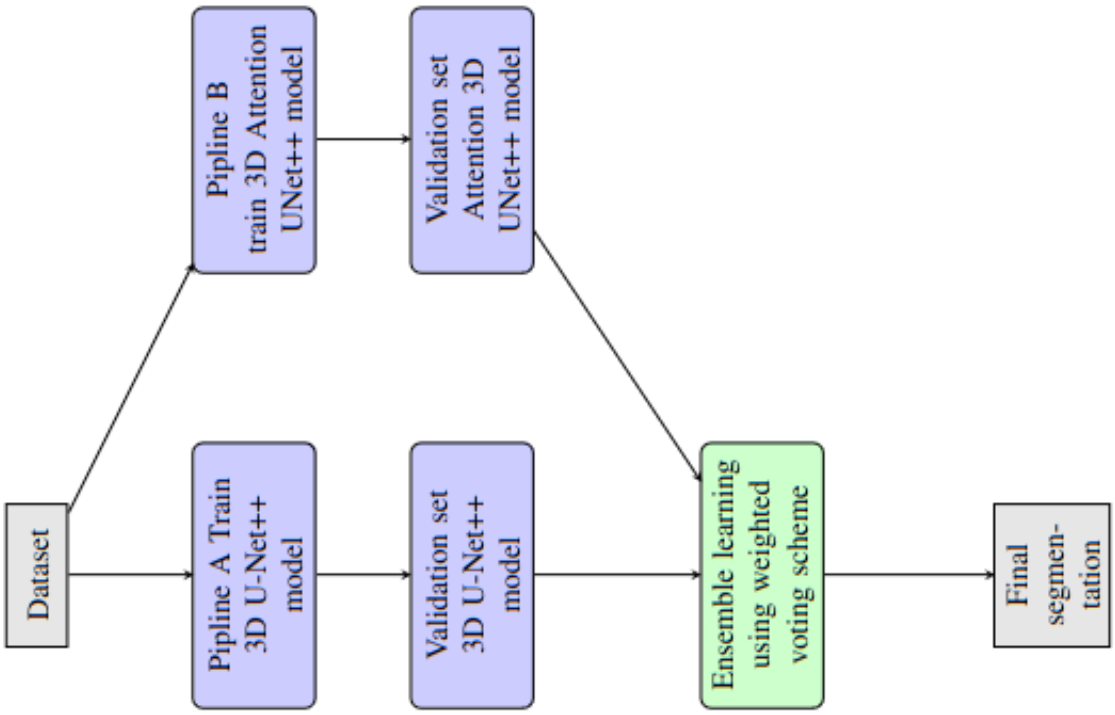


Fig 5: Strategy learning using Different Pipeline and Ensemble learning a weighted voting algorithm that used U-Net++ and Attention-U-Net++ pipelines

3.4 Training Details

in our study, we employed two widely used segmentation models,3D U-Net++ and 3D attention U-Net++ as explained in the early section, to segment Brain Tumor. Each model was trained independently with different hyperparameters and loss functions. To evaluate the models' performance, we used two standard metrics, Dice similarity coefficient (DSC) and Jaccard similarity coefficient (JSC), on a validation set.

3.4.1 loss Functions

During the ensemble training strategy, we separated each training process into individual pipelines, namely Pipeline A and B as Figure 5 illustrate, to describe how we trained each model. For the loss function, we ensured that each model was trained with a different loss function to capture the most features for segmentation and avoid loss of information in the spatial domain. Specifically, we used Binary Cross-Entropy (BCE) loss and Dice loss,[24] to form a new loss function called BCE-Dice loss in Pipeline B and Focal Tversky Loss[25] in Pipeline A.

(1) The BCE Loss: measures the difference between the predicted probability and the ground truth label. It is commonly used for binary classification tasks and is calculated by taking the negative logarithm of the predicted probability for the correct label as following Eq:5 .

(2) The Dice Loss: measures the overlap between the predicted segmentation and the ground truth segmentation. It is calculated by taking the ratio of twice the intersection of the two segmentations and the sum of the pixels in both segmentations as following Eq:6.

the BCE-Dice Loss is a combination of these two loss functions, the BCE-Dice Loss is able to capture both the spatial and label-wise information of the segmentation task. The BCE Loss encourages the model to correctly classify the pixels while the Dice Loss encourages the model to correctly segment the regions.

$$BCEDiceLoss = BCE + DiceLoss \tag{4}$$

where,

$$BCE(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N} y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i) \tag{5}$$

$$DiceLoss(y, \hat{y}) = 1 - \frac{2\sum_{i=1}^{N} y_i \hat{y}_i + \epsilon}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N} \hat{y}_i + \epsilon} \tag{6}$$

Here, y is the ground truth, $\hat{y}$ is the predicted output, and ϵ is a small value added to avoid division by zero errors.

The Focal Tversky loss is a modification of the Tversky loss that introduces a focusing parameter, γ, which is a hyperparameter that controls the degree of focusing. The Focal Tversky loss encourages the model to focus more on hard examples during training, which can improve its ability to segment challenging regions. Overall, the Focal Tversky loss is a useful loss function for image segmentation tasks, as it balances the need to accurately segment both foreground and background regions while also focusing the model's attention on hard examples . and defines as follow in Ep: 7

$$FocalTverskyLoss(y, \hat{y}) = (1 - Tversky(y, \hat{y}))^{\gamma} \tag{7}$$

(1) The Tversky loss: introduces an additional parameter to control the balance between false positives and false negatives during segmentation. The Tversky index is defined as the ratio of the intersection to the sum of the intersection and the union of the ground truth and predicted segmentation. The Tversky loss is then defined as 1 minus the Tversky index and can be used as a loss function to train segmentation models. is defined as :

$$Tversky(y, \hat{y}) = \frac{\sum_{i=1}^{N} y_i \hat{y}_i}{\sum_{i=1}^{N} y_i \hat{y}_i + \alpha \sum_{i=1}^{N} y_i (1 - \hat{y}_i) + (1 - \alpha) \sum_{i=1}^{N} (1 - y_i) \hat{y}_i} \tag{8}$$

where y is the ground truth, $\hat{y}$ is the predicted output, and α is a hyperparameter that controls the weight given to false positives and false negatives.

3.4.2 Evaluation metrics

To evaluate the performance of our segmentation models, we used two metrics: Dice Similarity Coefficient (DSC) and Jaccard Similarity Coefficient (JSC). These metrics are widely used for

image segmentation tasks and provide a measure of the similarity between the predicted and ground truth masks.

Dice Similarity Coefficient (DSC): The DSC measures the overlap between the predicted mask and the ground truth mask. It is defined as follows:

$$DSC = \frac{2 * |A \cap B|}{|A| + |B|} \tag{9}$$

where A is the predicted mask, and B is the ground truth mask.

Jaccard Similarity Coefficient (JSC): The JSC is another metric that measures the similarity between the predicted and ground truth masks. It is defined as follows:

$$JSC = \frac{|A \cap B|}{|A \cup B|} \tag{10}$$

where A is the predicted mask, and B is the ground truth mask.

4 Experiments and Results

To implement an ensemble learning strategy, we designed our training pipeline by running two separate pipelines, A and B, which are related to each of the two models we used, 3D U-Net++ and 3D Attention U-Net++. Both models were tested on the same validation set to ensure they were trained on the same sample size and track their performance. Finally, we combined the models using an ensemble learning weight voting algorithm to obtain the final segmentation. The entire implementation was done using Pytorch.[26] The code implementation can be found in our GitHub repository: https://github.com/deep-matter/AttentionUnetPlus_EnsembleLearning.

Pipline A In our first implementation in Pipeline A which includes the 3D U-Net-Plue-Plus model, we trained the model with various hyperparameters. The training was performed on a specific data split fold 0. As discussed earlier, the choice of loss function used during training can significantly impact the model's performance. Therefore, in this pipeline, we used the Focal-Tversky loss function. The table below describes the hyperparameters used in this pipeline A table 1.

Hyperparameters		Value	Description
fold	(N)	7	Number of folds to split the data to balance for StratifiedKFold parameter
Learning rate	(α)	3e-4	The step size for updating the model parameters Adam Optimizer.
Number of epochs	(N)	250	The number of times to iterate over the entire training dataset.
Accumulation step		4	The number of batches to accumulate gradients before updating the model.
Batch size		5	The number of samples to process in a single forward/backward pass.
Gamma	(γ)	0.5	Focusing parameter for Focal Tversky loss
Alpha	(α)	0.7	Balance between false positives and false negatives Focal Tversky loss parameter
input shape	$(\dim)$	(50, 50, 50)	Dimensions of input data
Output channel	(N)	(3, 50, 50, 50)	Number of output channels

Table 1: Hyperparameters used in the model training for Fold 0 , while Output channel in the first dimension refer to the number of classes

Pipline B as following the previous steps on Pipeline A we changed various hyper-parameters and loss functions BCE-Dice loss, the main idea behind training the model 3D attention U-Net++ in a different way is to make sure the model learns the most important features from data representation, over some experiments we did we found out number fold we give into StratifiedK Fold as parameter impact in a performance model which increases validation set, the data Per-processing does not change we stuck with the same process, and the table shows the update setup experiment of Pipeline B table 2.

Hyperparameters		Value	Description
fold	(N)	3	Number of folds to split the data to balance for StratifiedKFold parameter
Learning rate	(α)	2e-4	The step size for updating the model parameters Adam Optimizer.
Number of epochs	(N)	200	The number of times to iterate over the entire training dataset.
Accumulation step		4	The number of batches to accumulate gradients before updating the model.
Batch size		5	The number of samples to process in a single forward/backward pass.
input shape	$(\dim)$	(50, 50, 50)	Dimensions of input data
Output channel	(N)	(3, 50, 50, 50)	Number of output channels

Table 2: Hyper-parameters used in the model training for Fold 0 while Output channel in the first dimension refer to the number of classes

4.1 ablation Study for Pipeline A and B

Our study focused on evaluating the performance of two pipelines, A and B, on the BRAST 2020 dataset that was split into Training/Validation. Both pipelines used different models to train: Pipeline A used the 3D U-Net++ model, while Pipeline B used the 3D attention U-Net++ model. We found that the performance of both pipelines was significantly impacted by the selection of hyper-parameters and loss functions during training.

Figure 6 represents the Dice and Jaccard similarity scores for two pipelines, Pipeline A and Pipeline B, for three different regions of interest (ROIs): Whole Tumor (WT), Tumor Core (TC), and Enhancing Tumor (ET). The Dice similarity coefficient (DSC) is a measure of the overlap between the segmentation of the ground truth and the predicted segmentation, while the Jaccard similarity coefficient (JSC) is a measure of the agreement between the ground truth and predicted segmentation. For all three ROIs shown following Figure 7 that represent our segmentation results in both of Pipeline A and Pipeline B and Looking at the results in Table 3, we can see that Pipeline A achieved higher Dice and Jaccard Similarity scores for all tissue classes compared to Pipeline B. Specifically, the WT Dice score for Pipeline A was 0.88, compared to 0.82 for Pipeline B. The TC Dice score for Pipeline A was also higher at 0.87, compared to 0.82 for Pipeline B. Finally, the ET Dice score for Pipeline A was 0.73, compared to 0.69 for Pipeline B. These results suggest that the 3D U-Net++ model used in Pipeline A is more effective at differentiating tissue classes than the 3D attention U-Net++ model used in Pipeline B.

Based on the results obtained, Table 4 presents the confidence intervals for the evaluation metrics (Dice and Jaccard similarity) of two pipelines, namely Pipeline A and Pipeline B, for different classes (WT, TC, and ET).

The confidence intervals provide a range of values within which the true mean of the sample is likely to fall with a certain level of confidence. These intervals are calculated based on the sample

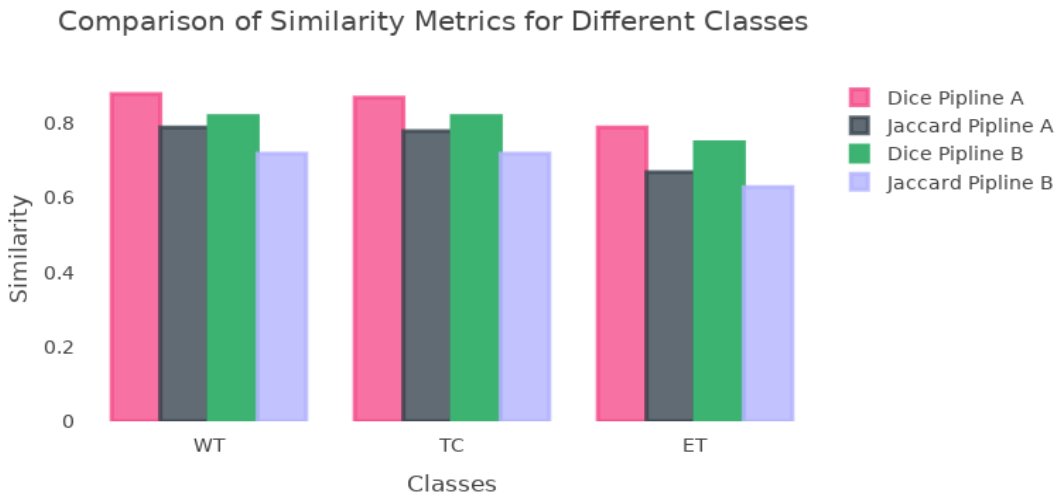
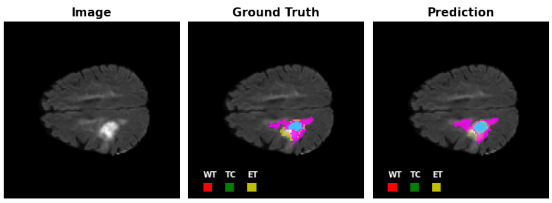


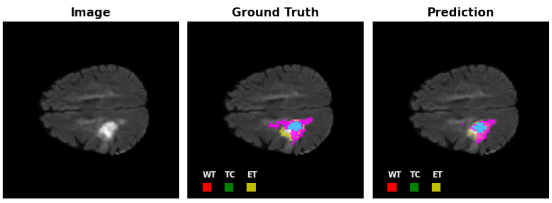
Fig 6: Comparison of Similarity Metrics for Different Classes

Pipeline	Dice Similarity			Jaccard Similarity		
	WT	TC	ET	WT	TC	ET
Pipeline A	0.88	0.87	0.73	0.79	0.78	0.59
Pipeline B	0.82	0.82	0.69	0.72	0.72	0.54

Table 3: Results of the evaluation metrics for two pipelines.



(a) Comparison of our segmentation results with Ground Truth labels Pipeline A



(b) Comparison of our segmentation results with Ground Truth labels Pipeline B

Fig 7: Comparison of our segmentation of all ROI results

Pipeline	Dice Similarity			Jaccard Similarity		
	WT	TC	ET	WT	TC	ET
Pipeline A	$(0.57, 1.18)$	$(0.57, 1.17)$	$(0.48, 0.99)$	$(0.51, 1.06)$	$(0.51, 1.05)$	$(0.38, 0.79)$
Pipeline B	$(0.78, 0.87)$	$(0.77, 0.88)$	$(0.64, 0.74)$	$(0.66, 0.77)$	$(0.66, 0.78)$	$(0.49, 0.59)$

Table 4: Confidence intervals (lower bound, upper bound) for the evaluation metrics of two pipelines.

statistics, including the class mean corresponding to each class prediction, class standard deviation corresponding to each class prediction, and the total sample size.

The formula to calculate the confidence intervals is as follows:

$$CI = \bar{x} \pm z \cdot \frac{s}{\sqrt{n}}$$

Where:
- CI is the confidence interval.
- $\bar{x}$ is the sample mean.
- z is the critical value corresponding to the desired confidence level.
- s is the sample standard deviation.
- n is the sample size.

By considering the confidence intervals, we can assess the precision and reliability of the sample estimates for each pipeline and class. Pipeline B generally has narrower confidence intervals compared to Pipeline A, indicating that Pipeline B shows more consistent and precise performance across the classes. However, further statistical analysis, such as hypothesis testing, would be necessary to determine if the differences between the pipelines are statistically significant.

Overall, the confidence intervals provide valuable information about the uncertainty associated with the performance metrics, allowing for a more comprehensive interpretation of the results and facilitating informed decision-making in selecting the most suitable pipeline for the task at hand.

4.2 ablation Study for Ensemble learning

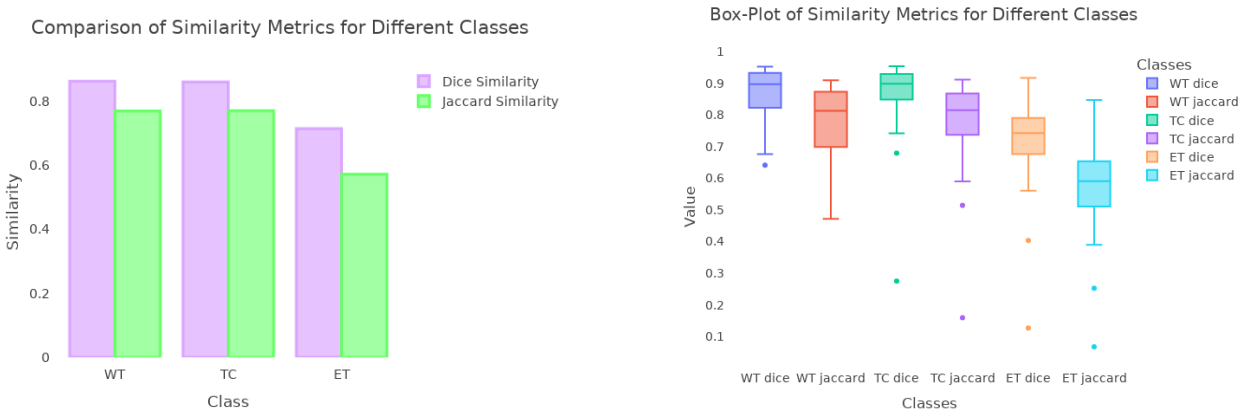
Our initial approach was to explore the use of ensemble learning with weighted voting to combine the outputs of both Pipelines A and B. The goal was to achieve more accurate results by leveraging the strengths of each individual model. To further improve the accuracy of the segmentation process, we proposed an ensemble learning approach that addressed algorithmic uncertainty in tumor segmentation using the BraTS 2020 dataset. The proposed method combined two different deep learning models with different architectures and loss functions, which were trained on the same subsets of the dataset. The final segmentation result was obtained through a weighted voting scheme, which allowed us to combine the predictions of the individual models in a way that maximized their accuracy. Based on the results presented in Table 5,

Models	Dice Similarity			Jaccard Similarity		
	WT Dice	TC Dice	ET Dice	WT Jaccard	TC Jaccard	ET Jaccard
Pipline A	0.88	0.87	0.73	0.79	0.78	0.59
Pipline B	0.82	0.82	0.69	0.72	0.72	0.54
Ensemble Learning	**0.86**	**0.86**	**0.71**	**0.77**	**0.77**	**0.57**

Table 5: Performance metrics for Pipeline A, Pipeline B, and Ensemble Learning on BRATS 2020 dataset

The performance metrics presented in Table 5 demonstrate the comparative results between Ensemble Learning and Pipeline B. A careful analysis of the table reveals that Ensemble Learning

consistently outperforms Pipeline B across all the evaluated metrics. In terms of Dice Similarity, Ensemble Learning achieves higher scores than Pipeline B for all three categories: WT Dice (0.86 vs. 0.82), TC Dice (0.86 vs. 0.82), and ET Dice (0.71 vs. 0.69). Similarly, when considering Jaccard Similarity, Ensemble Learning again surpasses Pipeline B in all three categories: WT Jaccard (0.77 vs. 0.72), TC Jaccard (0.77 vs. 0.72), and ET Jaccard (0.57 vs. 0.54). These results unequivocally demonstrate that Ensemble Learning enhances the overall performance Figure 8 when compared to Pipeline B,



(a) Comparison of Similarity Metrics for Different Classes

(b) Bax-Plot of similarity Metrics for Different Classes

Fig 8: Model performance on Validation Set

To compare the results among different learning strategies employed in our study, we conducted significance tests using the Dice coefficient and Jaccard index as similarity metrics. Table 6 presents the comparison of p-values obtained for all the models, highlighting the significance of choosing different strategies for training the model, including loss function selection and hyperparameter tuning. These comparisons are particularly relevant for 3D segmentation data.

Table 6: Summary of P-value Comparisons

Comparison	Dice Coefficient	Jaccard Index
Pipeline A vs. Pipeline B	0.0131	0.0091
Ensemble Learning vs. Pipeline B	0.0377	0.0229
Pipeline A vs. Ensemble Learning	0.0377	0.0377

1. Comparison: Pipeline A vs. Pipeline B

   (a) Dice Coefficient: The p-value for the dice coefficient between Pipeline A and Pipeline B is 0.0131, indicating a statistically significant difference. With a p-value below the commonly used significance level of 0.05, Pipeline A exhibits superior performance compared to Pipeline B in terms of the dice coefficient.

   (b) Jaccard Index: The p-value for the Jaccard index between Pipeline A and Pipeline B is 0.0091, which is also below 0.05. Therefore, there is a statistically significant difference, and Pipeline A outperforms Pipeline B in terms of the Jaccard index.

(c) Consequently, both the dice coefficient and Jaccard index provide evidence that Pipeline A has a statistically significant advantage over Pipeline B in terms of performance.

2. Comparison: Ensemble Learning vs. Pipeline B

   (a) Dice Coefficient: The p-value for the dice coefficient between the ensemble learning weight voting method and Pipeline B is 0.0377, indicating a statistically significant difference. The ensemble learning weight voting method demonstrates better performance compared to Pipeline B in terms of the dice coefficient.

   (b) Jaccard Index: The p-value for the Jaccard index between the ensemble learning weight voting method and Pipeline B is 0.0229, which is below the significance level of 0.05. Therefore, there is a statistically significant difference, and the ensemble learning weight voting method outperforms Pipeline B in terms of the Jaccard index.

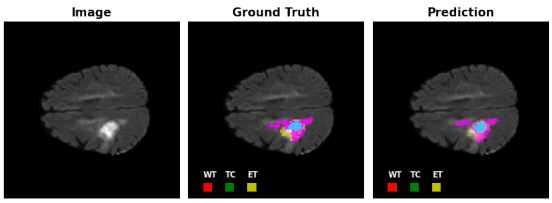
   (c) Thus, both the dice coefficient and Jaccard index suggest that the ensemble learning weight voting method has a statistically significant advantage over Pipeline B in terms of performance.

3. Comparison: Pipeline A vs. Ensemble Learning

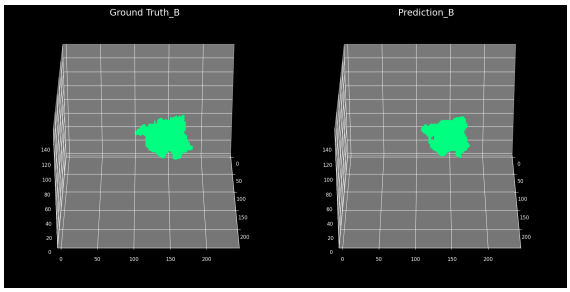
   (a) The p-value for both the Dice Coefficient and Jaccard Index comparisons is 0.0377, indicating a statistically significant difference between Pipeline A and the ensemble learning method in terms of both metrics.

Overall, the obtained results underscore the significance of selecting appropriate learning strategies as shown in Figure.9. including loss function and hyper-parameter tuning, in achieving improved performance for 3D segmentation data

Thus, it can be concluded that the ensemble learning approach significantly improves the prediction accuracy of the model over the Pipeline B approach. The segmentation results of all three regions of interest (WT, TC, and ET) are presented below in Figure 10.



(a) Comparison of our segmentation results with Ground Truth labels ensemble learning



(b) our segmentation results with Ground Truth labels:
3D extraction of ROI WT, TC, ET volumetric Shape The Whole Tumor

Fig 10: Comparison of our segmentation of all ROI results

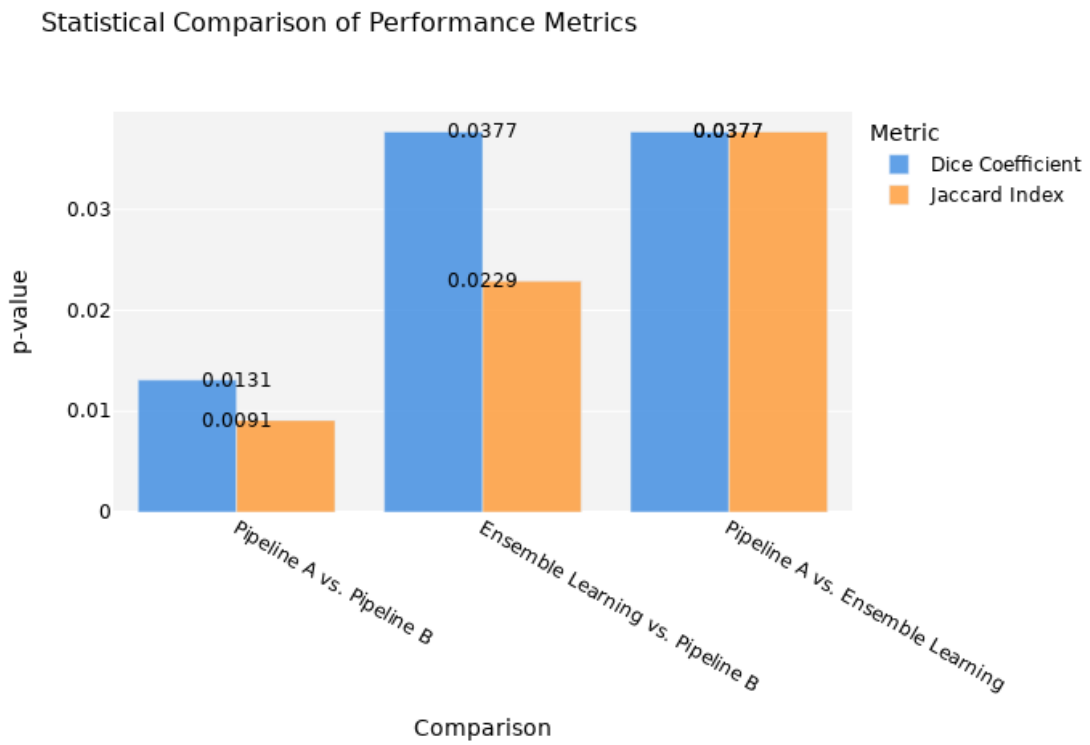


Fig 9: significance of selecting appropriate learning strategy based on P-Value

5 Discussion

In this study, we presented an intuition approach to glioma segmentation by utilizing the BraTS 2020 dataset. Our Strategies of learning approach by exploring different Pipelines, in our final stage we combine multiple deep learning models with different architectures and loss functions using a weighted voting scheme, outperforming Pipeline B in terms of both Dice Similarity and Jaccard Similarity coefficients. This success highlights the importance of addressing algorithmic uncertainty in the segmentation process.

However, our approach has some limitations. One major limitation is that we only used two deep learning models and we trained the model only of fold zero which that comes with resources Hardware limitation needs more GPUs, Moreover, Strategies of learning need time to scale and stable models during training, and future research may benefit from incorporating more models to improve performance. Another limitation is that our approach requires a significant amount of computational resources, which may be challenging to access in some settings.

Despite these limitations, our Strategies of learning approach represents a promising direction for future research in glioma segmentation. By improving the accuracy of segmentation using different model architectures and costume Loss functions, this approach may ultimately have a positive impact on clinical decision-making and patient outcomes.

6 Conclusion

In conclusion, we have demonstrated the potential of our proposed Strategies of learning approach for glioma segmentation. The combination of multiple deep learning models with different ar-

chitectures and loss functions and using Ensemble a weighted voting scheme offers a solution to address algorithmic uncertainty and improve the accuracy of segmentation. While further optimization and practical improvements are needed, this approach shows great promise for future applications in the field of glioma segmentation.

7 Disclosures

The authors declare that they have no conflict of interest

8 Acknowledgments

We would like to thank our respectful research assistant Moath Alawaqla, for his distinguished role of data collection.

9 Funding

This work is supported by Jordan University of Science and Technology, Irbid-Jordan,

References

1 E. C. Holland, "Progenitor cells and glioma formation," *Current opinion in neurology* **14**(6), 683–688 (2001).
2 H. Ohgaki and P. Kleihues, "Population-based studies on incidence, survival rates, and genetic alterations in astrocytic and oligodendroglial gliomas," *Journal of Neuropathology and Experimental Neurology* **64**(6), 479–489 (2005).
3 D. N. Louis, H. Ohgaki, O. D. Wiestler, *et al.*, "The 2007 who classification of tumours of the central nervous system," *Acta neuropathologica* **114**, 97–109 (2007).
4 G. P. Mazzara, R. P. Velthuizen, J. L. Pearlman, *et al.*, "Brain tumor target volume determination for radiation treatment planning through automated mri segmentation," *International Journal of Radiation Oncology* Biology* Physics* **59**(1), 300–312 (2004).
5 D. E. Morris, J. D. Bourland, J. G. Rosenman, *et al.*, "Three-dimensional conformal radiation treatment planning and delivery for low-and intermediate-grade gliomas," **11**(2), 124–137 (2001).
6 J. Hu, X. Gu, and X. Gu, "Mutual ensemble learning for brain tumor segmentation," *Neurocomputing* **504**, 68–81 (2022).
7 S. Das, S. Bose, G. K. Nayak, *et al.*, "Deep learning-based ensemble model for brain tumor segmentation using multi-parametric mr scans," *Open Computer Science)* **12**(1), 211–226 (2022).
8 M. Prastawa, E. Bullitt, S. Ho, *et al.*, "A brain tumor segmentation framework based on outlier detection," *Medical image analysis* **8**(3), 275–283 (2004).
9 A. Gooya, K. M. Pohl, M. Bilello, *et al.*, "Joint segmentation and deformable registration of brain scans guided by a tumor growth model," **14**(Pt 2), 532 (2011).
10 K. Kamnitsas, W. Bai, E. Ferrante, *et al.*, "Ensembles of multiple models and architectures for robust brain tumour segmentation," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: Third International Workshop, BrainLes 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 14, 2017, Revised Selected Papers 3*, 450–462, Springer (2018).
11 G. Wang, W. Li, S. Ourselin, *et al.*, "Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: Third International Workshop, BrainLes 2017, Held*

in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 14, 2017, Revised Selected Papers 3, 178–190, Springer (2018).
12 S. Bakas, H. Akbari, A. Sotiras, *et al.*, "Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features," *Scientific data* **4**(1), 1–13 (2017).
13 B. H. Menze, A. Jakab, S. Bauer, *et al.*, "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE transactions on medical imaging* **34**(10), 1993–2024, IEEE (2014).
14 S. Bakas, M. Reyes, A. Jakab, *et al.*, "Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge," *arXiv preprint arXiv:1811.02629* (2018).
15 S. Prusty, S. Patnaik, and S. K. Dash, "Skcv: Stratified k-fold cross-validation on ml classifiers for predicting cervical cancer," *Frontiers in Nanotechnology* **4**, 972421 (2022).
16 F. Pérez-García, R. Sparks, and S. Ourselin, "Torchio: a python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning," *Computer Methods and Programs in Biomedicine* , 106236 (2021).
17 K. Van Leemput, F. Maes, D. Vandermeulen, *et al.*, "Automated model-based tissue classification of mr images of the brain," *IEEE transactions on medical imaging* **18**(10), 897–908 (1999).
18 Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, *et al.*, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, 3–11, Springer (2018).
19 O. Oktay, J. Schlemper, L. L. Folgoc, *et al.*, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999* (2018).
20 O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, 234–241, Springer (2015).
21 R. Li, X. Wang, G. Huang, *et al.*, "A comprehensive review on deep supervision: Theories and applications," *arXiv preprint arXiv:2207.02376* (2022).
22 O. Oktay, J. Schlemper, L. L. Folgoc, *et al.*, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999* (2018).
23 X. Wang, R. Girshick, A. Gupta, *et al.*, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7794–7803 (2018).
24 F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 fourth international conference on 3D vision (3DV)*, 565–571, Ieee (2016).
25 N. Abraham and N. M. Khan, "A novel focal tversky loss function with improved attention u-net for lesion segmentation," in *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)*, 683–687, IEEE (2019).
26 A. Paszke, S. Gross, S. Chintala, *et al.*, "Automatic differentiation in pytorch," in *NIPS-W*, (2017).