GSNet: A Multi-class 3D Attention-based Hybrid Glioma Segmentation Network

Article *in* Optics Express · October 2023

DOI: 10.1364/OE.499054

Md Tasnim Jawad

Khulna University of Engineering and Technology

10 PUBLICATIONS **232** CITATIONS

Ashfak Yeafi

Khulna University of Engineering and Technology

4 PUBLICATIONS **2** CITATIONS

Optics EXPRESS

GSNet: a multi-class 3D attention-based hybrid glioma segmentation network

MD TASNIM JAWAD,[*] ASHFAK YEAFI, AND KALYAN KUMAR HALDER

Department of Electrical and Electronic Engineering, Khulna University of Engineering & Technology, Khulna 9203, Bangladesh
[*]*mdjawad006@gmail.com*

Abstract: In modern neuro-oncology, computer-aided biomedical image retrieval (CBIR) tools have recently gained significant popularity due to their quick and easy usage and high-performance capability. However, designing such an automated tool remains challenging because of the lack of balanced resources and inconsistent spatial texture. Like in many other fields of diagnosis, brain tumor (glioma) extraction has posed a challenge to the research community. In this article, we proposed a fully developed robust segmentation network called GSNet for the purpose of glioma segmentation. Unlike conventional 2-dimensional structures, GSNet directly deals with 3-dimensional (3D) data while utilizing attention-based skip links. The network is trained and validated using the BraTS 2020 dataset and further trained with BraTS 2019 and BraTS 2018 datasets for comparison. While utilizing the BraTS 2020 dataset, our 3D network achieved an overall dice similarity coefficient of 0.9239, 0.9103, and 0.8139, respectively for whole tumor, tumor core, and enhancing tumor classes. Our model produces significantly high scores across all occasions and is capable of dealing with newer data, despite training with imbalanced datasets. In comparison to other articles, our model outperforms some of the state-of-the-art scores designating it to be suitable as a reliable CBIR tool for necessary medical usage.

1. Introduction

1.1. Problem presentation

Glioma, a form of primary brain cancer, is caused by an abnormal proliferation of glial cells located in the cerebrum or the cerebellum. It is a tumor of the malignant category and can be life-threatening if left undiagnosed and untreated. In a 5-year span, the rate of recovery from brain tumors for individuals under the age of 15 is around 75% whereas, for people aged 15 to 39, the rate is 72% [1]. In 2020, around 308,102 individuals were identified as having a primary central nervous system (CNS) tumor [1]. This year, an estimated 24,810 individuals from the US will be diagnosed with primary malignant CNS tumors [2].

Segmentation is a preliminary technique for early detection and enhanced treatment options. It involves dividing an image into regions and assigning each of them a label or category based on its semantic meaning [3]. Glioma segmentation, like that of many other tumors, is a critical step in treatment planning which makes it possible to precisely assess and quantify tumor shape, size, and location, as well as to differentiate between various types of tissue. Despite the benefits of glioma segmentation, the procedure can pose challenges due to the following reasons [4]:

- On medical images, glioma can appear heterogeneous, with a variety of intensities and textures across separate territories of the tumor. This makes discriminating the tumor from normal brain tissue difficult.
- Glioma can differ in terms of location and size, making it difficult to accurately segment them. Large tumors might be difficult to identify from surrounding healthy brain tissue,

https://doi.org/10.1364/OE.499054
Received 27 Jun 2023; revised 13 Oct 2023; accepted 21 Oct 2023; published 19 Nov 2023

whereas small tumors may be obscured by image noise or overlap with other structures in the brain.

- Glioma segmentation can be a time-consuming and labor-intensive process, particularly when carried out manually by a radiologist or medical expert. Access to annotated medical images is critical to the precision of glioma segmentation algorithms.

Automated methods based on computer-aided biomedical image retrieval (CBIR) tools can be used to speed up and increase the efficiency of the segmentation process. The construction of these tools depends on computational capability. However, once properly constructed, the CBIR tool can aid in general usage irrespective of consumer hardware specifications. Section 1.2 discusses some of these computerized methods in more detail.

1.2. Literature review

Computerized segmentation strategies have gained increasing attention in recent years. Despite the difficulties and restrictions associated with traditional segmentation techniques, researchers have persisted in creating new strategies that are intended to generate precise and trustworthy results [5]. This section highlights some of the recent articles that demonstrated different types of computerized segmentation techniques.

A combination of the whale optimization algorithm and fuzzy c-means clustering to achieve optimal segmentation results is proposed by [6]. The authors, along with magnetic resonance imaging (MRI) images, used synthetic samples to validate their proposed system's performance. They showed the effectiveness of their proposed strategy by introducing artificial noise in their image samples. Their approach gained a minimum mean squared error of 28.36, 27.26, and 25.09 at 7%, 5%, and 3% exposure to salt and pepper noise. An alternate hyper-heuristic-based approach that combines the benefits of various algorithms and techniques to deliver improved results is proposed by [7]. This method takes into account multiple objectives and constraints to produce a more accurate and efficient segmentation outcome. The authors addressed the limitations of meta-heuristic-based approaches to segmentation and divided their work into two parts. Firstly, they used the genetic algorithm (GA) to define the order of the meta-heuristic approach. Then they deployed the meta-heuristic-based approach based on the sequence achieved from the GA. The authors compared their approach based on various performance metrics, namely, the structural similarity index, CPU time(s), peak signal-to-noise ratio (PSNR), and fitness functions. They conducted their experiment against a collection of original and hybrid meta-heuristic approaches and found promising scores. A modified watershed segmentation (MWS) algorithm was proposed by the authors of [8] to improve upon traditional watershed segmentation. They incorporated modifications that enhanced the algorithm's ability to accurately segment brain regions of interest (ROIs). Before applying MWS, with the help of a Xilinx Virtex-5 FPGA, they processed their corresponding MRI images through a high-pass filter. This process removed the low-pitched noises and retained the high-pitched details. They also applied the enhanced canny edge detection algorithm to identify the boundaries of the brain regions, gaining 99.31% accuracy. Another wavelet transform-based image segmentation method is proposed by [9], where the authors identified the threshold level for segmentation via valley point. The valley point threshold level selection aids in locating the best value for segmentation, which most accurately distinguishes the target object from the background. The authors compared their experiments with conventional segmentation methods namely, the maximum variance segmentation method, the bimodal segmentation method, and the valley threshold segmentation method. In addition to SNR and PSNR, they included the processing time and the value of threshold as evaluation metrics and showed their model's overall performance accuracy.

In recent years, a significant number of neural network (NN)-based segmentation approaches were seen in various articles [3]. These encoder-decoder-based segmentation networks have gained popularity due to their capacity to handle complicated structures, process massive amounts of data, and shorten the time needed for pre-processing and attribute extraction. Segmentation networks are also capable of learning from the input and extracting features, which makes them perfect for coping with novel datasets and ambiguous structures.

One of the oldest and most common forms of segmentation networks is called UNet, which was first published as a research article in [10]. It contains a contracting path as a feature extractor and an expansive path as a feature generator in which the information is downscaled and upscaled, respectively. The downsampler usually referred to as the encoding path, uses several convolution and max-pooling layers to increase the quota of attribute maps while decreasing the overall spatial resolution of the input image. The upsampler, often referred to as the decoding path, is made up of several transposed convolution layers that lessen the quota of attribute maps while increasing the resolution. The upsampler's fine details and the downsampler's contextual information are combined to create a precise segmentation using the UNet architecture [10]. A variation of the UNet-like structure has been implemented by various researchers in different articles where they proposed their version of the model and modified it for specific tasks. One such example is seen in [11], where the authors presented a 2-dimensional (2D) convolutional neural network (CNN) based ensemble of networks for brain tumor segmentation. Initially, the authors extracted the whole tumor from the background using three segmentation networks. Then they processed their results using Growcut which is a cellular automaton-based seed-growing technique. Lastly, they extracted the sub-regions using more ensemble of networks. They obtained a range of $0.74 - 0.85$ dice similarity coefficient (DSC) accuracy which was comparable to the other proposed approaches. They were unable to conduct experiments on 3-dimensional (3D) networks due to the high computational cost.

Several articles used various types of NN blocks to improve segmentation accuracy. One such example can be found in [12], where the authors used the pre-trained ResNet model in modifying their UNet's encoder [13]. The authors proposed a shared encoder-based structure, followed by separate decoders for separate classes. They received favorable DSC and demonstrated some complications throughout their experiment. They also stated that their model appeared unsuitable for practical use due to its 3D size. To address some of the limitations of conventional UNet, the authors of [14] proposed Sharp UNet, a novel UNet-type segmentation network. The Sharp UNet improves on the traditional skip connection-based architecture by incorporating a depthwise convolution of the attribute extracted map along with a sharpening kernel filter, making the network more spatially sensitive and capable of recognizing minute details in the input image. Having tested their architecture on six different datasets, the authors obtained excellent validation results. Their model outperformed some of the most advanced baseline structures. Another version of the UNet-like model is seen in [15]. In it, the authors used bottleneck residual blocks and also provided significant details regarding the fine-tuning of their proposed network. They compared their work with a very well-known segmentation network called DeepLabV3+. Despite the 2D nature of their network, it still achieved mean DSC of 0.8673, 0.7514, and 0.7983 on three separate classes of their utilized dataset. Due to a greater number of enhancing areas, the authors faced some difficulties and the model did not perform well across all target classes. A similar but 3D approach was seen in [16] and [17] where the authors proposed their UNet-type networks which were capable of dealing with 3D segments. Each voxel in the volume is classified as belonging to the item of interest or the background using the features that the 3D CNN layers of the networks learn to extract from the given data. In the case of [17], the topology of the segmentation network was more hybridized, with numerous learning modules cascaded one after the other to boost overall efficiency. A fundamentally distinct yet simple version of the segmentation network was proposed by [18], where the network itself was an improvement of the well-known VNet model [19].

The studies discussed above show that deep learning-based segmentation models are critical for tumor segmentation. When applied to the dataset utilized in this article, CNN-based segmentation architectures have produced promising results for glioma segmentation [20]. Furthermore, mathematical or algorithmic methods frequently lack the ability to re-train or learn newer data through adaptation, making validation for the segmentation task less reliable. We, therefore, aim to contribute by proposing a new CNN-based network for extracting ROIs from brain segments. Our focus will be on glioma segmentation, which accounts for a significant portion (33%) [21] of all cancers.

2. Our contribution

We have developed a hybrid end-to-end multi-class 3D CNN-based segmentation network named GSNet (glioma segmentation network) for automating the segmentation process. Our network consists of 5 levels in the encoder and 5 levels in the decoder, forming a 5-stage segmentation network incorporating attention-based skip connections. Our simulation results demonstrate a significant improvement, with GSNet performing on par with, if not surpassing, some of the top-performing segmentation networks developed for glioma segmentation. The overall contribution of our work is stated below:

- Creating an end-to-end segmentation network (GSNet) capable of segmenting glioma segments with high efficiency.
- Developing a robust 3D network capable of pooling both low-level and high-level attributes from MRIs.
- Producing an overall lightweight model, that is able to perform quick segmentation while dealing directly with 3D images, despite training with significantly imbalanced data.
- Achieving high accuracy without utilizing extensive image pre-processing. Obtaining performance scores that are large enough to establish GSNet as a state-of-the-art model.
- Compressing the entire pipeline into a user-friendly GUI application, which can serve as an automated tool enabling practitioners and medical personnel to efficiently segment glioma ROIs.

The remaining sections are presented as follows:

Section 3 deals with the datasets and the methodologies utilized to carry out the experimentations. The simulation results are mentioned in section 4. This section is elaborated into multiple segments representing each dataset and also contains a comparison with other well-known methods involving the primary dataset. The article is finalized in section 5 with the conclusion.

3. Materials and methods

3.1. Datasets

We trained our model, GSNet, using the multimodal brain tumor image segmentation benchmark (BraTS) datasets [22]. We primarily used the BraTS 2020 for both training and validation. We also compared the performance of GSNet through BraTS 2019 and BraTS 2018 datasets, using which we further trained and validated our model. Each dataset comprises MRI scans of brain segments (glioma) with a voxel size of $1 \times 1 \times 1$ mm^3 and an original volume shape of $240 \times 240 \times 155$ dimension. The datasets include T1-weighted (t1), T1-weighted contrast-enhanced (t1ce), and T2-weighted (t2) scans, as well as fluid attenuation inversion recovery (FLAIR) images. The datasets also contain annotated ground truth values all of which are labeled using four values (0, 1, 2, 4), where 0 denotes no tumor and the rest belong to different regions. For segmentation, our segment classes were tumor core (TC) (label 1,4), whole tumor (WT) (label 1-4), and enhancing tumor core (ET) (label 4) [23]. Here, the WT regions represent the entirety of the infected portion including the peritumoral edema part and the TC regions represent the more aggressive parts. The ET regions highlight a subset of the core portion which indicates increased cell density [22]. However, the class distribution of the datasets was imbalanced. Each sample contains four images and a segmentation mask file, stored in the ".nii" file format. The datasets were extracted from various clinical protocols covering 19 institutions and the images inside were segmented manually and were also approved by experts [23]. The process was laborious and the slices were also carefully revised for proper intensity, texture, and various other morphological parameters [22]. To construct GSNet, for training, we used 176 samples from the "BraTS2020_TrainingData" folder, which is less than the total of 369 samples in the BraTS 2020 dataset. The selection of the train-test split was initially done at random with a 5-fold stratified k-fold approach [24]. Among the 5, one of the folds with 44 samples was kept separate for validation. This entire distribution was later kept fixed for the remainder of our experiments. The BraTS 2019 dataset consists of 259 samples of high-grade glioma (HGG) patients' MRI scans and 76 samples of low-grade glioma (LGG) patients' MRI scans. The BraTS 2018 dataset contains 210 HGG and 75 LGG patients' MRI scans. We only used the HGG patients' MRI samples from the 2019 and 2018 datasets.

3.2. Data pre-processing

To enhance the learning capability of our network and reduce computational costs, we utilized center cropping on all images. Figure 1 represents an implementation of the center cropping procedure which involved selecting the middle region of each image as the training feed while ignoring the outlying portions. This method focuses on the ROIs within the image, as it is often the case that the areas surrounding the object of interest contain irrelevant information that could negatively impact the network's performance. In the case of our utilized data, the MRI scans along with the tumor and its corresponding regions are all in the center. This is why with the help of cropping, the network can learn only the essential features, leading to improved segmentation accuracy [25]. We resized the images from their original size to 128 × 128 × 128 resolution, resulting in a total of 2,097,152 pixels in each voxel image. This size strikes a balance between reducing the computational burden and preserving important image details. Additionally, the process helped us to train our network with limited GPU resources. The right-most column in Fig. 1 is displaying the ground truth for all three classes (WT: cyan, TC: navy blue, ET: magenta). Both the images and their masks are cropped in the same way.

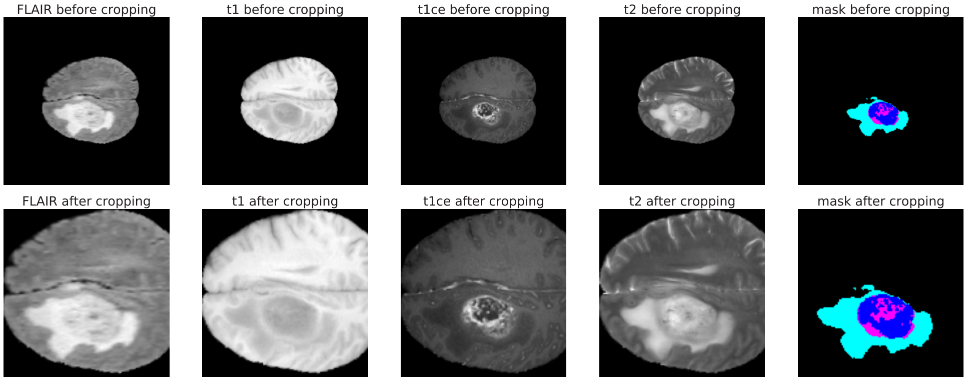


Fig. 1. The 100th slice of id 'BraTS20_Training_140'. The top row is displaying the original slices and the bottom row is displaying the center-cropped versions.

3.3. Methodology

3.3.1. Network fundamentals

The feature learning process of our model, GSNet, is expanded using 3D-CNN, which consists of both depth and spatial dimensions. Equation (1) gives the general formula for 3D CNN which represents the calculation method of the position (x, y, z) in the i^{th} convolutional layer of the j^{th} feature cube [26].

$$Y_{i,j}^{x,y,z} = f\left(\sum_{m}\sum_{l=0}^{L_i-1}\sum_{w=0}^{W_i-1}\sum_{c=0}^{C_i-1} k_{i,j,m}^{l,w,c} Y_{(i-1),m}^{(x+l),(y+w),(z+c)} + b_{i,j}\right) \tag{1}$$

Here, let the input kernel number, data shape, and convolution kernel size in the 3D-CNN-based operation be n, $D \times L \times N \times C$, and $q \times q \times q$, respectively where D and L represent the width and height of tensor, q denotes the coverage of the convolution kernel over spectral dimension in each convolution operation, N is the band number and C is the channel number. Here, the dimension of the feature map is $(D - q + 1)(L - q + 1)(N - q + 1)n$ created by the 3D convolutional if padding is not used and the stride is kept 1. Because of the extra dimension, the attribute map generated by the equation contains more spectral information. In addition, 3D convolutional layers have more parameters than 2D ones, enabling more intricate and nuanced reconstructions of the input data [27]. We used padding to keep the input and output dimensions constant while using multi-scale 3D convolution kernels. The procedural outputs along the spectral dimension are concatenated to create the output attribute map of the first layer. The attributes are then subjected to a number of additional convolutional layers for extra processing in order to integrate and abstract them. We have employed a particular pooling procedure called 3D max pooling (MaxPool3d) [28]. In the simplest case, if the function's input size is (B, C, D, L, W), the output size is $(B, C, D_{out}, L_{out}, W_{out})$, and the kernel size is (pD, pL, pW) then, for input $(B_i, C_j, stride[0] \times d + p, stride[1] \times l + m, stride[2] \times w + n)$ the output can be precisely calculated from Eq. (2).

$$\operatorname{out}\left(B_i, C_j, d, l, w\right) = \max_{p=0,\ldots,pD-1} \max_{m=0,\ldots,pL-1} \max_{n=0,\ldots,pW-1} \tag{2}$$

This layer helps our network to separate the input feature maps from a series of non-overlapping regions and their corresponding maximum value from each zone. It also decreases the geometrical size of the attribute maps and lowers the number of involved parameters and associated computational expenses.

We utilized 3D upsampling which is a type of layer used in CNN-based models for upsampling an input tensor [29]. The nearest-neighbor resampling technique is at the heart of the upsample formula used in our work which involves increasing the size of the input tensor by a specified factor along each dimension, by duplicating the values of the nearest neighboring pixel (Eq. (3)). We employed bilinear interpolation in the upsampling layer.

For an input (N, C, P_{in}), (N, C, L_{in}, P_{in}) or $(N, C, D_{in}, L_{in}, P_{in})$ the corresponding output (N, C, P_{out}), $(N, C, L_{\text{out}}, P_{\text{out}})$ or $(N, C, D_{\text{out}}, L_{\text{out}}, P_{\text{out}})$, where,

$$\begin{aligned} D_{\text{out}} &= \lfloor D_{\text{in}} \times \text{scale \_factor} \rfloor, \\ L_{\text{out}} &= \lfloor L_{\text{in}} \times \text{scale \_factor} \rfloor, \\ P_{\text{out}} &= \lfloor P_{\text{in}} \times \text{scale \_factor} \rfloor \end{aligned} \tag{3}$$

We also used instance normalization (InstanceNorm3D), a deep-learning normalization approach, for regularization. InstanceNorm3D can be used to normalize activations of a mini-batch so that each example has a zero mean and unit variance (Eq. (4)) [30]. In contrast to batch normalization, which performs normalization across the entire batch, the InstanceNorm3D layer normalizes each instance in a mini-batch resulting in every instance having a distinct mean and variance, which aids in the prevention of covariate shifts, reducing overfitting and allowing for more stable internal activations [30].

$$y = \gamma * \frac{x - Me[x]}{\sqrt{Var[x] + \epsilon}} + \beta \tag{4}$$

Here x is the input tensor, $Me[x]$ is the mean value of the batch, $Var[x]$ is the variance calculated from the batch, y is the output tensor, γ, and β are learnable scalings and shift parameters, and ϵ is a small constant added to prevent division by zero.

Skip-connection is a design pattern used in deep neural networks to address the vanishing gradient problem [31]. The objective behind such a pattern is to cut through some of the network's layers, giving the model more direct access to previous features and enabling gradients to flow back into the input through backpropagation more quickly and effectively [10]. In other words, the fine-grained attributes learned in prior layers are kept in the skip connections that are established between non-adjacent levels, allowing data to be transmitted from one layer to the other with ease.

We have employed an attention mechanism to construct our skip connections [32,33] which was originally made by the article [34]. The process forms a query and a set of key-value pairs to an output, which is calculated as a weighted sum of values. The weighting factors are measured using the attention mechanism which forms the compatibility metric between the query and its corresponding key. Our implementation of an attention-based skip link can be visualized in short from Fig. 2. In our network, the inputs from both the encoder side and decoder side were fed directly into the attention block which as shown in Fig. 2, processed the inputs and created an output which was later concatenated with the next upcoming convolution block from the decoder. The following equations describe the working principle of attention layers. Basically, if the encoder creates N number of the hidden state vectors where every one of them has dimension r, then the input shape of the feedforward layer is *(N, 2r)*. While being added with a bias term B, this input I is multiplied with a matrix W which has the shape *(2r, 1)*. This subsequently produces the score S where this output has the dimension *(N, 1)* (Eq. (5)).

$$S = I[N * 2r] * W[2r * 1] + B[N * 1] \tag{5}$$

The score (S), is then fed through a *tanh* function which is later followed by a *softmax* activation function (Eq. (6)) to get the normalized alignment scores.

$$A = softmax(tanh(S)) \tag{6}$$

Finally, the result A is multiplied by one of the input terms *(IT)* (Eq. (7)).

$$C = IT * A \tag{7}$$

We follow this mechanism to connect our encoder to our decoder as skip links. The feature maps from each level are weighted by the attention mechanism according to their importance. The weighted features are then employed in the skip connection to send data from an early layer in the network to a later layer. This allows our network to focus on the most crucial elements from the feature map. Overall it helps improve the segmentation accuracy.

To carry out the construction of the segmentation model, two different activation functions, namely the rectified linear unit (ReLU) and the softmax functions were used in addition to the Adamax optimizer. The ReLU activation function is a piecewise linear process that induces non-linearity into the network [35]. The ReLU function is defined as follows:

$$f(q) = max(0, q) \tag{8}$$

Here, q is the input to the ReLU activation function which returns the given value if it is positive, and zero if it is negative. Its use is supported by its simplicity, computational efficiency,

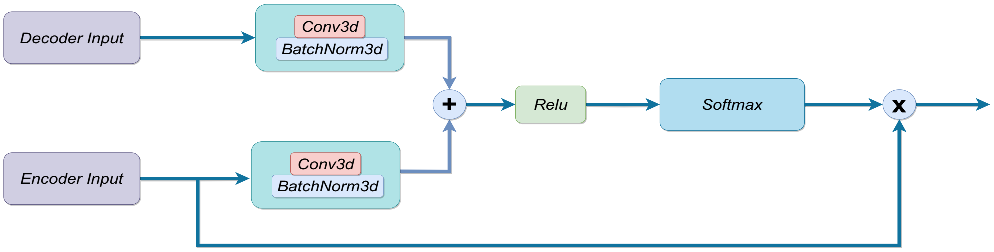


Fig. 2. Attention mechanism used in GSNet for the implementation of skip connection. The figure highlights the internals of the main network's attention block.

and good performance on a wide range of tasks. The softmax activation function [36] is a popular function used in machine learning, especially in multi-class classification problems. It takes in a vector of real numbers and creates a probability distribution that is ranged among Q classes. The softmax function is defined mathematically as follows:

$$y_i = \frac{e^{(z_i)}}{\sum_{j=1}^{Q} e^{(z_j)}} \tag{9}$$

where z_i is the i^{th} element of the input vector z, e is the base of the natural logarithm, and the denominator guarantees that the sum of the probabilities is 1.

3.3.2. Evaluation metrics

The dice similarity coefficient (DSC) score is basically a similarity metric used to evaluate the performance of image segmentation algorithms. As shown in Eq. (10), to measure DSC, first the intersection of the ground truth data and the segmented image is multiplied by 2 then, the result is divided by the addition of their sizes [37].

$$DSC(p, \hat{p}) = \frac{2|p \cap \hat{p}|}{|p| + |\hat{p}|} \tag{10}$$

$$IoU(y, \hat{p}) = \frac{|p \cap \hat{p}|}{|p \cup \hat{p}|} \tag{11}$$

The Jaccard index or intersection over union (IoU) score, is also a similarity metric used to compare the diversity of sample sets. Like DSC, the intersection of the aforementioned sets is used, but in this case it is divided directly by their union (Eq. (11)) [37]. For both Eqs. (10) and (11), the ground truth segmentation is p, and the predicted segmentation is $\hat{p}$. Both the DSC and IoU scores range from 0 (complete mismatch) to 1 (perfect match).

The dice loss (DL) is a loss function calculated based on the DSC formula and is widely used in machine learning, particularly in image segmentation tasks, to measure the dissimilarity between predicted segmentation maps and true segmentation maps [38]. The formula for calculating the DL is given below (Eq. (12)).

$$DL(p, \hat{p}) = 1 - \frac{2\hat{p} + 1}{p + \hat{p} + 1} \tag{12}$$

$$FL(\hat{p}) = -\alpha(1 - \hat{p})^{\gamma} * log(\hat{p}) \tag{13}$$

where the ground truth segmentation is p and the predicted segmentation is $\hat{p}$. Here, 1 is added in both the numerator and the denominator to ensure that the formula is not undefined in edge case scenarios such as when $p = \hat{p} = 0$. The focal loss (FL) is a loss function that was initially introduced for classification tasks but can also be applied to segmentation tasks where the class imbalance is a common issue [39]. In the context of segmentation tasks, the primary objective is to label each pixel or voxel in an image with a corresponding class label, although, some classes may have a small number of examples, leading to class imbalance. FL addresses this issue by assigning higher weights to the minority class examples and reducing the contribution of well-classified examples to the loss function. Equation (13) highlights the formula for calculating FL, where $\hat{p}$ is the predicted probability of the correct class, α is a weighting factor that gives more importance to the minority class, and γ is a focusing parameter [39]. Sensitivity (Sen) as shown in Eq. (14), is a metric commonly used to evaluate the performance of image segmentation algorithms which measures the proportion of true positive pixels that are correctly identified. Sen is defined as the ratio of true positives (TP) to the sum of TP and false negatives (FN) [40].

$$Sen = \frac{TP}{TP + FN} \tag{14}$$

A high Sen score indicates that the algorithm is good at detecting objects in the image, while a low Sen score means that many object pixels were missed or incorrectly identified as background.

3.3.3. Proposed GSNet model

Throughout this section, we highlight the constructional features of our segmentation network. We have devised a straightforward input-to-output workflow (see Fig. 3) that incorporates consistent pre-processing techniques, outlined in section 3.2. At the core of this pipeline is our proposed GSNet model, which manipulates the inputs and generates masks for ROI extraction. The network comprises several sequential convolutional blocks that have been carefully trained and optimized for heightened precision. Additionally, attention-based skip links have been integrated into the model to enhance contextual understanding between the internal layers.

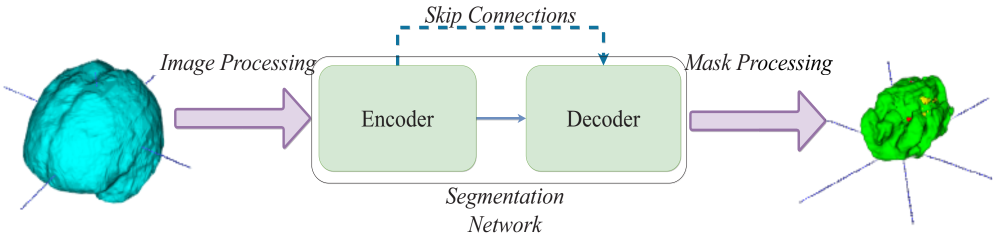


Fig. 3. An end-to-end encoder-decoder-based concept which takes the entire 3D image for segmentation.

Traditionally, a segmentation network contains two separate parts, namely, the encoder and the decoder where the encoder obtains high-level attributes from the given images. It typically consists of multiple convolutional layers with pooling techniques to reduce the spatial range of the input data while increasing the number of attribute channels. The output of each layer can then be passed through an activation function to induce non-linearity in the model. The final output of the encoder is a set of activation maps that learns high-level semantic information about the input image [41]. The decoder is a part of the network that upsamples the feature maps with low resolution from the encoder to the original input resolution by utilizing a series of up-convolutions or transposed convolutions that gradually increase the spatial resolution of the feature maps. The final output of the decoder is a probability map that assigns a class label to each pixel in the input data, representing the predicted segmentation mask [41].

For our GSNet model, we have proposed a 5-level encoder and a corresponding 5-level decoder, resulting in a 5-stage segmentation network. Our encoder's core is made up of several repeating convolutional blocks each of which contains a 3D convolutional layer (Conv3d), a 3D instance normalization layer (InstanceNorm3d), a 3D dropout layer (Dropout3d), a ReLU activation layer, in that order (Fig. 4(left)). Because this sequence is repeated twice in each of these blocks, we simply refer to it as Conv 2.

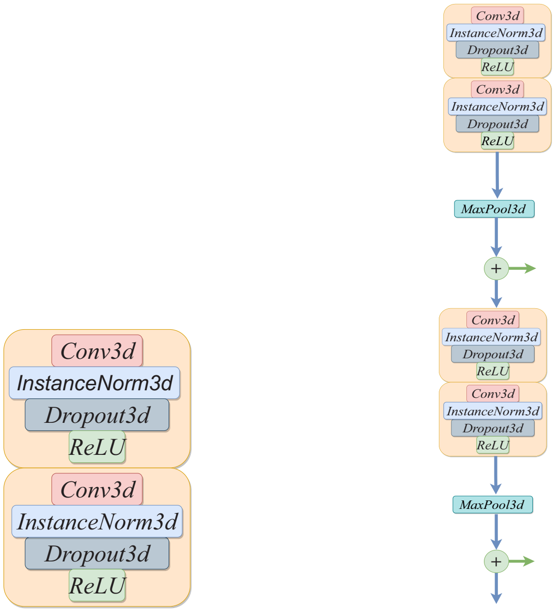


Fig. 4. (left) Unit convolutional block of the encoder (Conv 2). (right) An example of a sequential 2-level encoder network comprised of Conv 2 unit and MaxPool3d layer.

Each level of the encoder is separated using a MaxPool3d layer with a kernel size of 2×2×2 and a stride of 2 × 2 × 2 (Fig. 4(right)). This layer is downsampling each batch by a factor of 2 and the entire arrangement of blocks is built in a way that facilitates the extraction of higher-level features and increases the network's overall accuracy. The overall sequential multilevel combination shown in Fig. 4(right), is maintained throughout the entire GSNet's encoder. The topological parameters involved throughout the first block of the encoder is summarized in Table 1.

Table 1. The properties of the first encoder block of GSNet's downsampler

Conv3d	*Input Channel* *4*	*Output Channel* *16*	*Kernel Size* (3 × 3 × 3)
InstanceNorm3d	*Feature Number 16*		
Dropout3d	*Probability 0.2*		
ReLU			
Conv3d	*Input Channel* *16*	*Output Channel* *16*	*Kernel Size* (3 × 3 × 3)
InstanceNorm3d	*Feature Number 16*		
Dropout3d	*Probability 0.2*		
ReLU			

The table highlights some of the key parameters, for instance, the convolution layers, besides using a kernel with the size of 3 × 3 × 3 are also utilizing a stride of 1 × 1 × 1, with the additional bias term, B continuously ignored throughout the learning phase. This slight adjustment is beneficial to keep our network unbiased toward the training data. The entire table represents an encoder level, where each level is basically scaled down by a factor of 2, and the channel numbers are continuously multiplied by 2. For example, the final layer of this entire block creates an output of 16 channels. Consequently, the next block contains 16-channel input and 32-channel output. The final output of the encoder from the corresponding level (5th block) creates a 256-channel output.

The decoder unit (Up Conv Block) of GSNet features an upsampling layer, followed by a Conv3d layer, an InstanceNorm3d layer, and a ReLU activation function in that order (Fig. 5(left)). To enable an upsampling procedure that adds context from higher levels, these blocks are stacked one on top of the other where the upsampling layer scales the data by a factor of 2, and the 'Nearest Neighbour' mode is maintained in all of them. The functionality of these layers is mentioned in section 3.3.1.

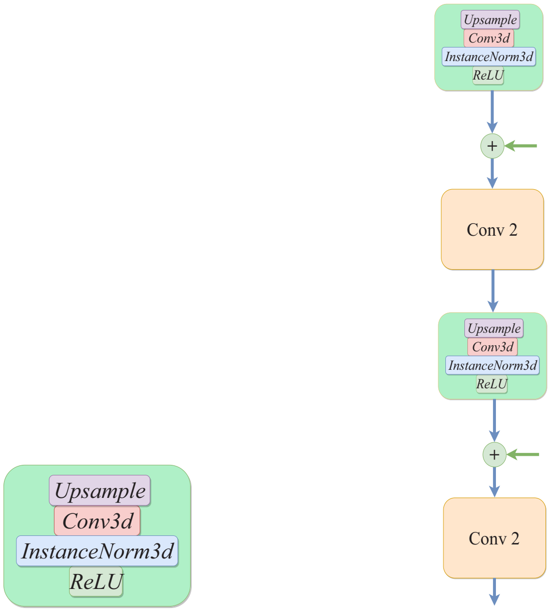


Fig. 5. (left) Decoder unit (Up Conv Block) for the purpose of upsampling. (right) An example of a sequential 2-level decoder network with Conv 2 blocks.

The Up Conv Block is followed by a Conv 2 unit and this whole sequence is repeated multiple times (around 5 times) to obtain a higher level of features and reconstruct the corresponding ROIs. The sequence of the decoder is built based on multiple training and testing and continuous error analysis and it follows the structure displayed in Fig. 5(right) where the additional Conv 2 blocks are regularizing the attribute maps. The green arrows are obtained from the encoder as they take in extra features and are being concatenated to the upcoming Conv 2 blocks on the decoder side. The first input channel in the decoder is 256 as it comes directly from the encoder and it is continuously downscaled by a factor of 2. Similar to that of the encoder, Table 2 represents the properties of the last decoder block of our proposed GSNet model.

The absolute encoder is serially connected to the decoder, and the connection between the levels is accomplished via attention-based skip connections (section 3.3.1 and Fig. 2). The final structure of GSNet is shown in Fig. 6.

Table 2. The properties of the final decoder block of the GSNet's upsampler

Upsample	*Scale Factor 2.0*	*Mode = 'nearest'*	
Conv3d	*Input Channel 32*	*Output Channel 16*	*Kernel Size* $(3 \times 3 \times 3)$
InstanceNorm3d	*Feature Number 16*		
ReLU			
Conv3d	*Input Channel 32*	*Output Channel 16*	*Kernel Size* $(3 \times 3 \times 3)$
InstanceNorm3d	*Feature Number 16*		
Dropout3d	*Probability 0.2*		
ReLU			
Conv3d	*Input Channel 32*	*Output Channel 16*	*Kernel Size* $(3 \times 3 \times 3)$
InstanceNorm3d	*Feature Number 16*		
Dropout3d	*Probability 0.2*		
ReLU			

GSNet, with the help of skip connections, learns both low-level and high-level attributes. Traditional skip-connection methods often suffer from a lack of optimization and some of the over-scaled skip-connection methods can also introduce gradient explosion [42]. The attention mechanism allows the decoder network to access information from the encoder network that is relevant to the upsampled features, which results in a more accurate and detailed segmentation mask. Besides, the use of skip links helps to mitigate the issue of vanishing gradients, as it enables gradients to flow directly from the decoder to the encoder network [43]. The network contains a total of 6493291 parameters. The modification and building procedures are based primarily on the DSC Score, along with IoU Score, DL, and FL where the outputs were continuously visualized after each iteration. We have provided the source code for our segmentation model at the link [44].

3.4. Hardware and training protocol

The training procedure utilizes the Python (version 3.7.6) programming language and the Pytorch framework (version 1.6.0). More specifically, the protocol involves the 'torch' module and is run on a cloud-based Ubuntu 16.04 operating system with 12GB RAM. The kernel utilizes 2 CPU cores and the Nvidia P100 GPU with 16GB GPU memory. GSNet is trained with a 0.0005 learning rate and the Adamax optimizer [45]. Similar to the selection process of constructing the network, the choice of the optimizer is also done based on a few trials where Adamax performs better. The remaining parameters involving the optimizer are kept in their default states and no further fine-tuning is done. We also choose to train with the specified low learning rate to avoid the problem of local minima [46]. The simulation procedure is trained and validated for 250 epochs. After 250 epochs, the model tends to overfit significantly which is why we kept the training limited to 250 epochs. The following sections highlight the simulation results.

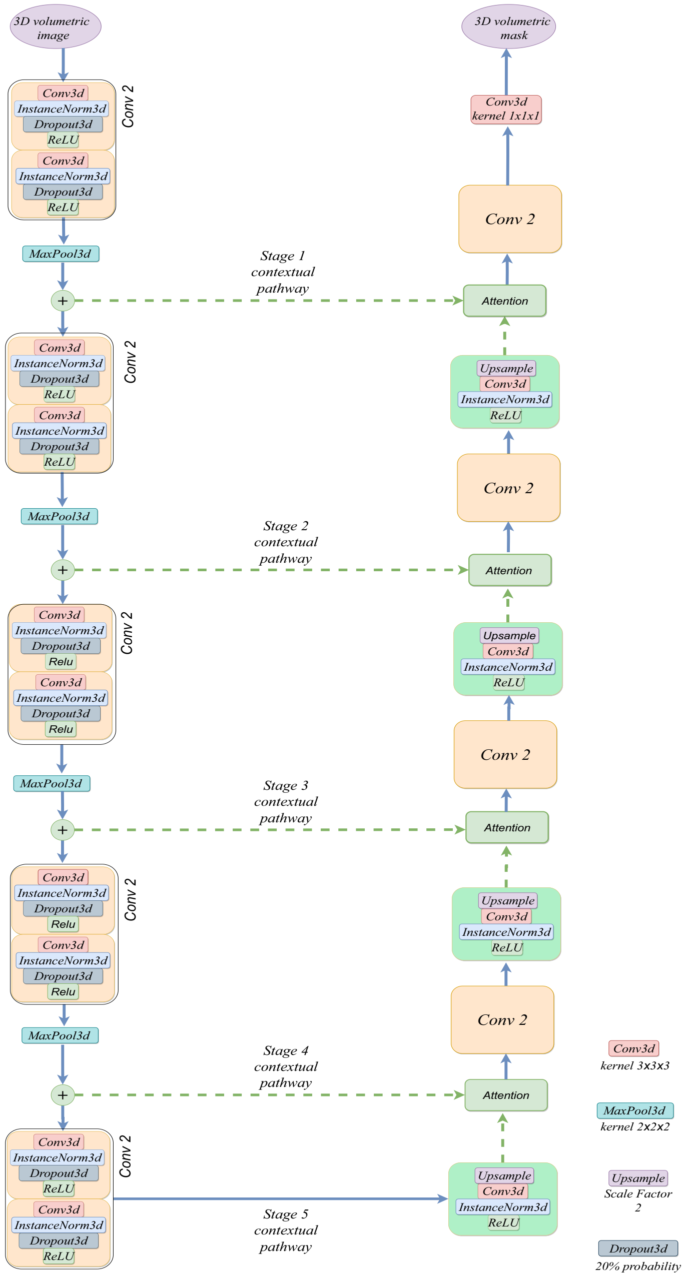


Fig. 6. The proposed GSNet structure for glioma segmentation.

4. Results and discussion

4.1. BraTS 2020

As mentioned earlier in section 3.1, we have used the BraTS 2020 dataset to construct our network, GSNet. Although only half of it was utilized for training, the dataset also served the purpose of validation. Due to the imbalance among segment classes, avoiding a high overfit was a prevailing challenge. Based on extensive experimentation, we have determined that the optimal

UNet-type structure is produced by having an equal number of decoder units corresponding to the number of encoder units. The skip connections were also modified based on the contextual levels of the encoder-to-decoder relationship, with the most accurate results being obtained from a symmetrical encoder-to-decoder connection, and at each level of the encoder, an output was generated and fed directly to an attention unit. An example of some of our earlier experiments is shown where a simple 3-stage encoder-decoder structure (3-stage skeleton) (Fig. 7) was trained for 50 epochs.

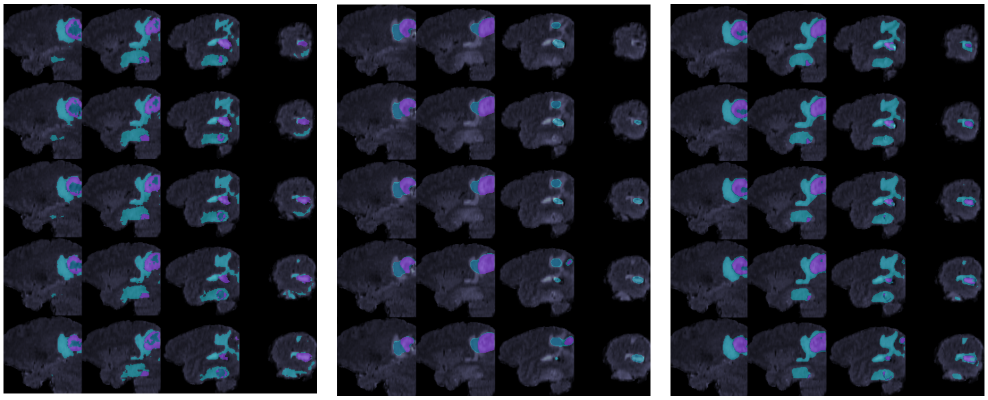


Fig. 7. A comparison of original ground truth and predicted segmentation mask (id BraTS20_Training_029), here the marks are predicted using a basic 3-stage encoder-decoder skeleton. Starting from the left, the first image batch (left) shows the original ground truth, the batch in the middle (middle) shows the predicted outcomes using the 3-stage skeleton without skip connections, the batch at the right (right) shows the predicted outcomes using the skeleton with skip connections. (WT: cyan, TC: navy blue, ET: magenta)

As we can see from Fig. 7, each batch of 20 images contains multiple slices of one single 3D MRI and the usage of attention-based skip connections provides a much better segmentation outcome. Analyzing the slices, we can observe that the outcome using the 3-stage skeleton with skip-connections Fig. 7(right) is much more similar to that of the original ground truths Fig. 7(left). Now the batch produced using the 3-stage skeleton without skip connection is shown in Fig. 7(middle). It is obvious that the results are poorly produced and do not reflect the original slices as in most of them, the classes are partially originated. Furthermore, the segmentation results obtained exhibit an incomplete rendering of the target regions, with areas that have not been fully or accurately delineated. Figure 8 shows the DSC calculated for all 3 classes (WT, TC, ET) separately after utilizing the 3-stage skeleton. Although for the class WT, the scores are pretty similar, the usage of skip connections shows a significant difference among the TC and ET classes. A 6% improvement in producing the TC regions and similarly, 5% improvement in producing the ET regions can be observed due to the usage of skip connections. This is why it is safe to say that our attention-based skip connections are making our GSNet more accurate.

We further experimented on the size of our network's structure a highlight of which is summarized in Table 3 where we measured the effectiveness of different contextual pathway levels and found that a 3-stage structure lacked the ability to learn detailed attribute maps. Increasing the overall levels (7-stage structure) improved the learning capability but the results were prone to overfitting due to the excessive use of Conv3d layers. As seen from the table, the gap between training and validation metrics is significantly high, and the training DSC is almost 19% higher than that of the validation DSC. After multiple modifications, we determined that a 5-stage contextual pathway structure is the most suitable, as it does not overfit to a significant

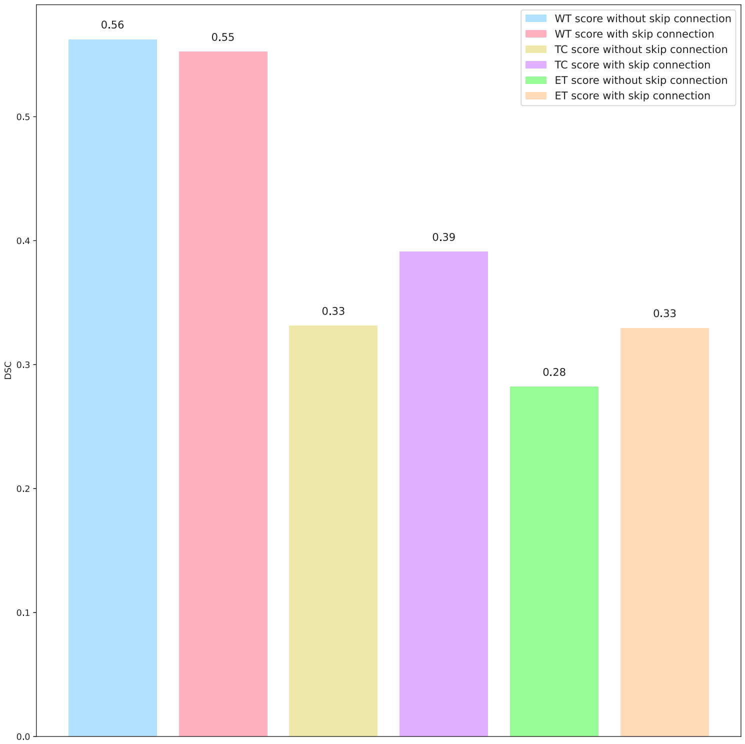


Fig. 8. Coefficient score comparison between the usage of a 3-stage skeleton network with skip connection and without skip connection.

extent and greatly improves validation accuracy. The final version of our model produces an initially under-fitted score more of which is displayed in Fig. 9 and Fig. 10.

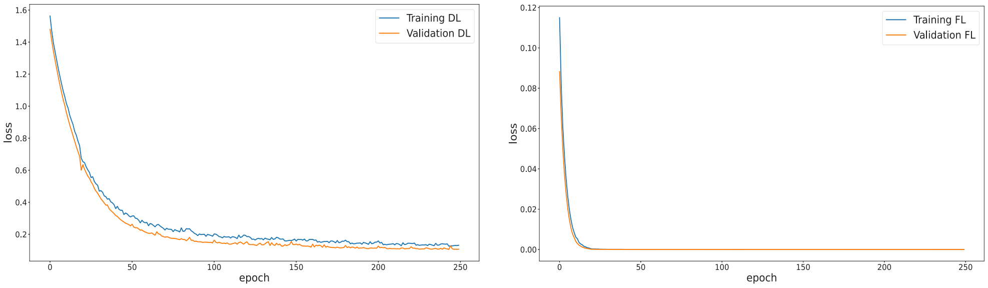


Fig. 9. The loss values obtained from training our proposed GSNet model for 250 epochs. (left: DL, right: FL)

Figure 9 represents the corresponding loss curves from training our model with the BraTS 2020 dataset for 250 epochs. The corresponding coefficient scores are visualized in Fig. 10. By utilizing DSC, we can effectively handle the discrepancy between the foreground and background [47], and with the help of FL, it is easier to focus on subsets of hard examples as it applies an additional modulating term to deal with class imbalance [39]. As observed from Fig. 9(left), the network produces a slightly lower validation loss compared to the training loss suggesting that the model may be underfitting, which is a positive indication. The focal loss in terms of training and validation is almost identical which reflects a proper learning pattern.

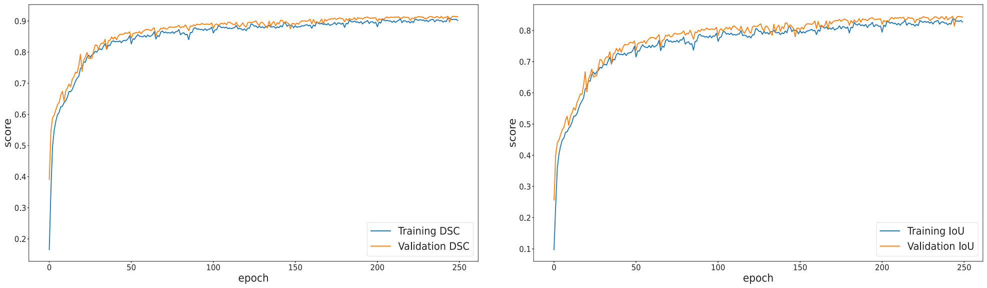


Fig. 10. The evaluating scores obtained from training our proposed GSNet model for 250 epochs. (left: DSC, right: IoU)

Table 3. Comparison of loss values and coefficient scores between 3-stage, 5-stage, and 7-stage networks after 50 epochs of training

	Training DL	Validation DL	Training DSC	Validation DSC
3-stage architecture	1.0724	1.2134	0.7014	0.5130
5-stage architecture	0.3085	0.2526	0.8376	0.8615
7-stage architecture	0.8716	0.9269	0.7906	0.5690

The coefficient scores reflect the robustness of the GSNet in accurately segmenting the target areas, demonstrating its superiority in this task. In the case of both DSC and IoU scores, the results convey a similar outcome where in some epochs, the training scores are lower and the validation scores are higher indicating a case of underfit. Again, throughout the entire training period, both the training and validation curves are very similar where the underfit is more visible in the case of the DSC curve.

After training for 250 epochs, our proposed GSNet scores around 0.90 in terms of DSC and over 0.80 in terms of IoU, both for the case of validation. For a more detailed analysis, we calculated the individual performance scores for all 3 classes which are visualized in Fig. 11. Generally, the class with the biggest amount often tends to create a biased network, but in our case, although WT does carry better performance than the rest, overall both for TC and ET the scores are promising. Besides the DSC and IoU, we have also measured the Sen scores among all three classes. Here, as we can observe from Fig. 11, despite training with an imbalanced dataset, both the DSC and IoU scores are significantly high. The first 3 bar from the left side, represents the WT region segmentation scores one of which is around 0.9239 DSC reflecting a good segmentation performance. Additionally, the 0.86 IoU and the 0.91 Sen score indicate a good overlap with a low false negative rate, which means that GSNet is less likely to not segment a pixel belonging to the WT region. For both TC and ET, the Sen scores are pretty high, while the IoU scores are relatively average. Regardless, the DSC scores across all three classes are impressive, as GSNet produces 0.9103 for TC and 0.8139 for ET region segmentation. The Sen scores across all 3 regions (WT, TC, ET) indicate a high recall count ensuring the model's capacity to identify the relevant pixel. These coefficient values indicate our network's supremacy.

Here, the BraTS 2020 dataset contains a very low amount of ET regions among its MRI images, which is the likely reason for the performance score to be relatively lower than that of the other two regions. Even then, the 0.8139 score is significant and offers decent reliability. We like to conclude that all of these bar plots are measured from a portion of the dataset kept separate

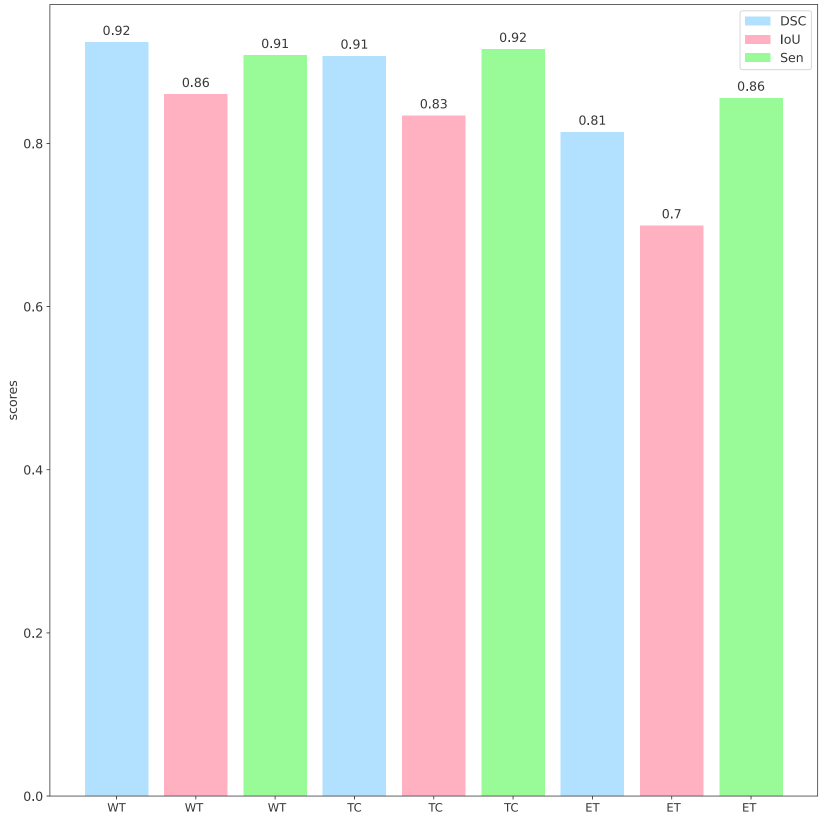


Fig. 11. Validation coefficient scores obtained for each separate class after 250 epochs of training (BraTS 2020 dataset).

from the training (as mentioned in section 3.1), so all these coefficient scores are from validation, which again, indicates that our proposed network is capable of dealing with newer unseen data.

A few examples from BraTS 2020 dataset are shown below (Figs. 12,13, and 14) where the masks are obtained using our proposed GSNet model. Here each batch of 20 images represents multiple slices of one single MRI image, the corresponding Id of which is given on the left side of each comparison. The true segmented masks are given in the dataset, which we are comparing against the predicted outcome from the GSNet model. The similarity between both batches is notice worthy as the predicted outcome almost creates a replica of the original segmentation ground truth. The shapes of the corresponding classes are pretty well constructed. In terms of the edges, especially in the WT region, all are well-bounded with not many mismatches or fragmented artifacts. This is a very desirable outcome conveying a very accurate segmentation.

We have further fine-tuned and accurately optimized the parameters of the network to achieve efficient glioma segmentation and formed a lightweight structure that is reliable, and suited perfectly for CBIR implementation. To validate the capability of our network, we conducted comparative tests with other datasets while maintaining a constant learning rate and optimizer (Adamax) (as mentioned in section 3.4) throughout the entire training process. Our results demonstrate extensively high efficiency across all 3 datasets and the model produces an adequate validation score even with the presence of imbalanced data [48] (mentioned in section 3.1). The network performs relatively well for all classes and originates an accurate semantic segmentation outcome that closely matches the original ground truth data provided by the datasets.

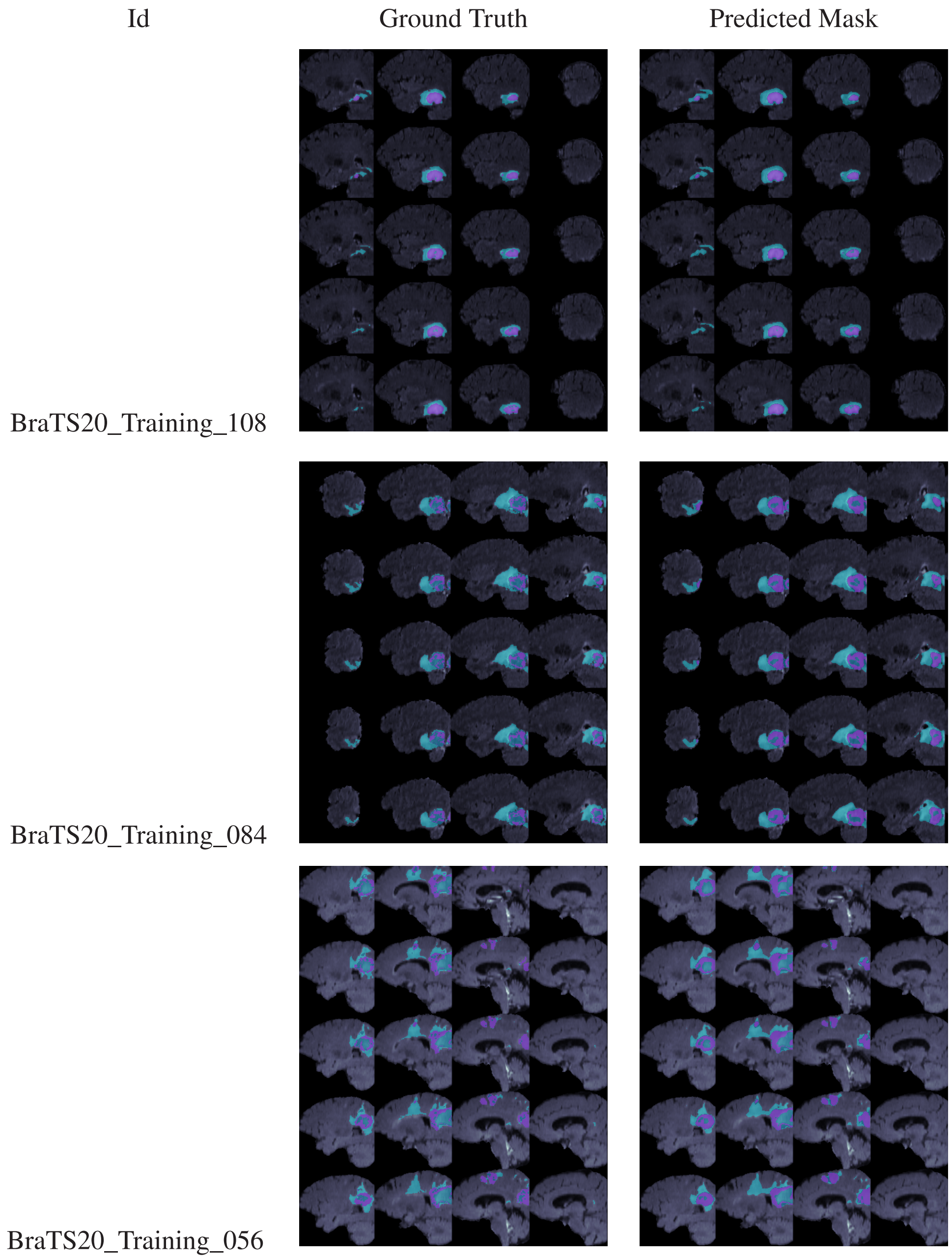


Fig. 12. The predicted mask obtained through the GSNet model in comparison with the given ground truth. (WT: cyan, TC: navy blue, ET: magenta), (Patient Id: BraTS20_Training_108, BraTS20_Training_084, BraTS20_Training_056)

4.2. BraTS 2019 and BraTS 2018

Figures 15 and 16 represent the overall learning performance of GSNet while training with BraTS 2019 and BraTS 2018 datasets accordingly. Here, as seen from both of these figures, the training curves and the validation curves reflect a similar scenario to that of the curves of BraTS 2020 mentioned in the previous section.

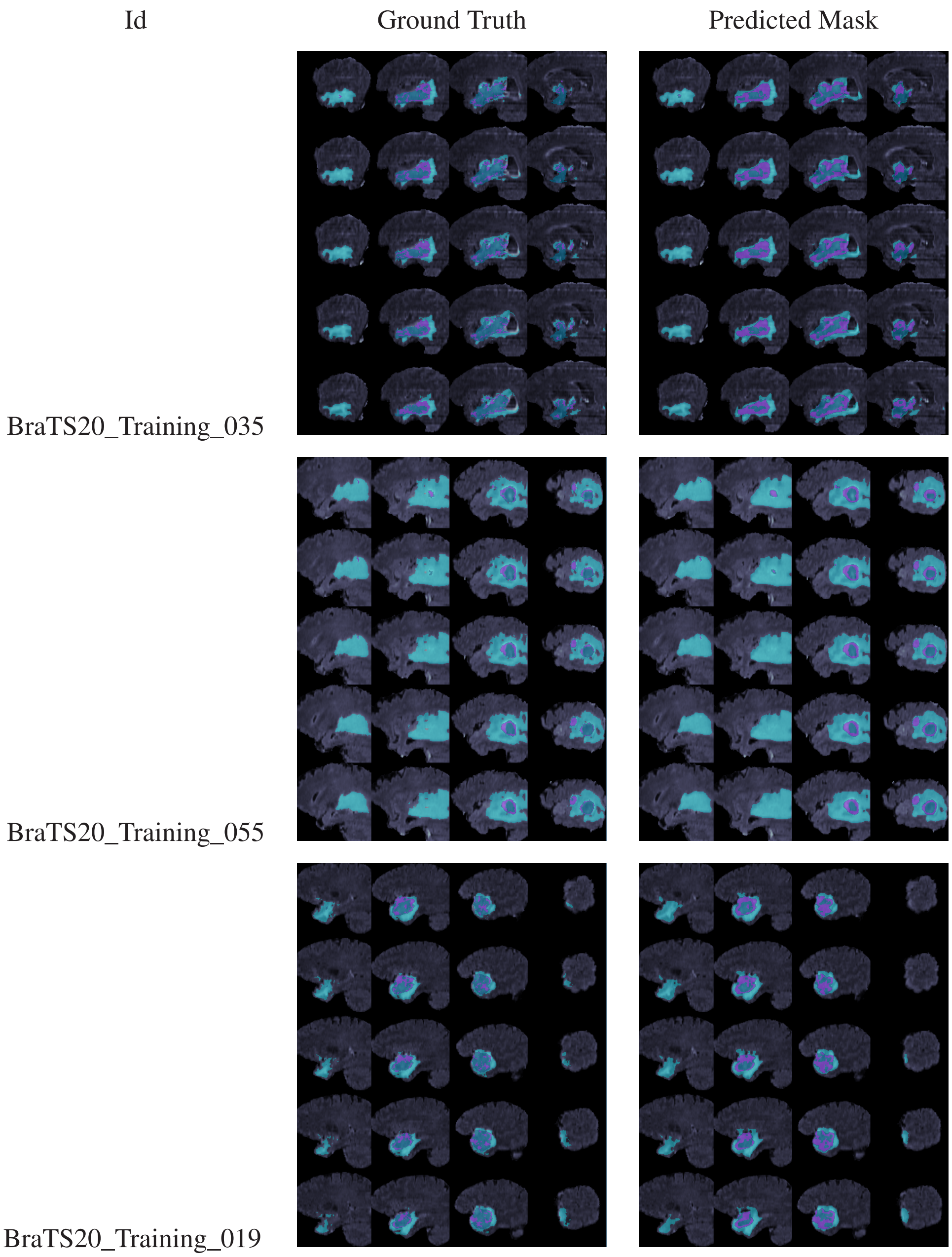


Fig. 13. The predicted mask obtained through the GSNet model in comparison with the given ground truth. (WT: cyan, TC: navy blue, ET: magenta), (Patient Id: BraTS20_Training_035, BraTS20_Training_055, BraTS20_Training_019)

Despite constructing our model with BraTS 2020, both for BraTS 2019 and BraTS 2018, GSNet scored significantly good results. For both of these datasets, around 0.90 DSC scores are produced using GSNet, and the loss values are also reasonably low. So even though these are different datasets, the model can learn the proper attributes if trained with GSNet, and like BraTS 2020, the validation scores are also significantly high. Table 4 represents the individual

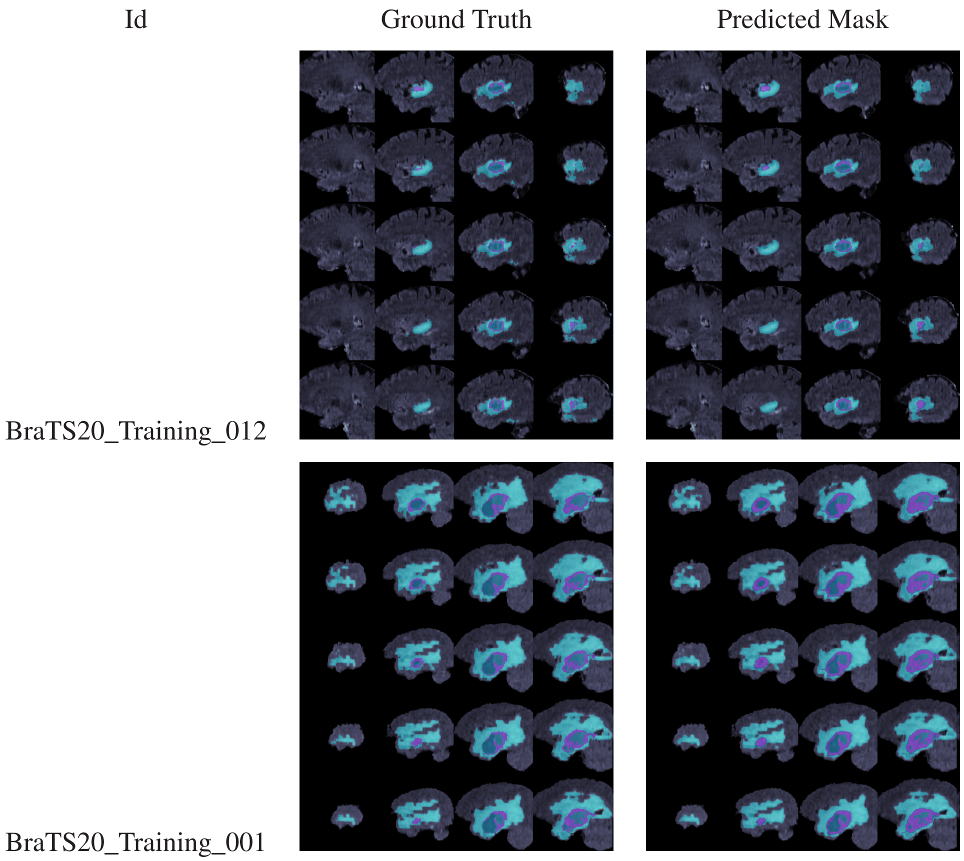


Fig. 14. The predicted mask obtained through the GSNet model in comparison with the given ground truth. (WT: cyan, TC: navy blue, ET: magenta), (Patient Id: BraTS20_Training_012, BraTS20_Training_001)

coefficient scores for all three classes and it highlights the overall validation accuracy. As seen from the table, similar to that of BraTS 2020, the DSC scores of WT, TC, and, ET in both BraTS 2019 and BraTS 2018 are respectively 0.8977, 0.8698, 0.7907, and 0.9048, 0.8759, 0.7956.

Table 4. Coefficient scores obtained for each separate class after training with BraTS 2019 and BraTS 2018 datasets

		WT	TC	ET
BraTS 2019	DSC	0.8977	0.8698	0.7907
	IoU	0.8197	0.7820	0.6704
	Sen	0.9207	0.9288	0.8818
BraTS 2018	DSC	0.9048	0.8759	0.7956
	IoU	0.8286	0.7877	0.6747
	Sen	0.8824	0.8708	0.8580

Compared to WT and TC, the ET segmentation scores are low as expected from the imbalanced datasets, a similar case to that of BraTS 2020. These scores validate that our network is very capable of training with newer data and obtaining very high accuracy. In terms of Sen, the BraTS

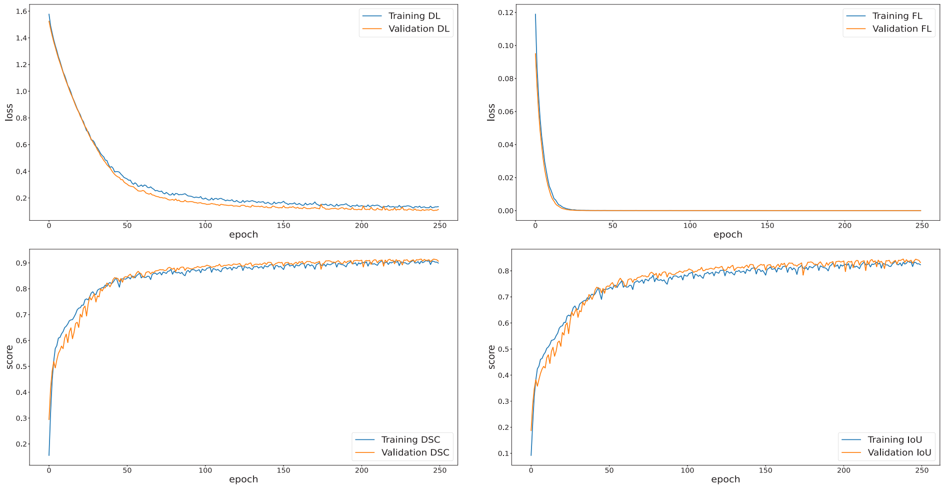


Fig. 15. The results obtained from training our proposed GSNet model for 250 epochs using the BraTS 2019 dataset. (Training curve: blue, Validation curve: orange) (top left: DL, top right FL, bottom left: DSC, bottom right: IoU)

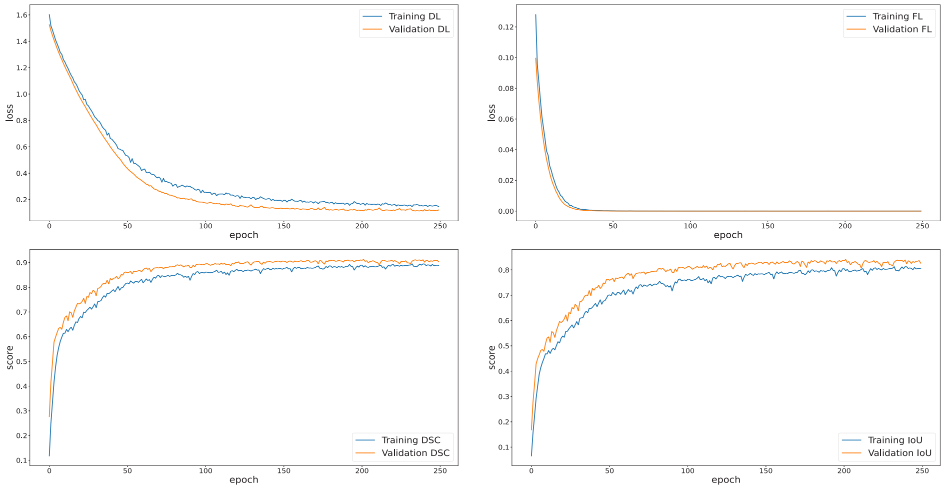


Fig. 16. The results obtained from training our proposed GSNet model for 250 epochs using the BraTS 2018 dataset. (Training curve: blue, Validation curve: orange) (top left: DL, top right FL, bottom left: DSC, bottom right: IoU)

2019 dataset shows greater performance whereas in terms of DSC, the BraTS 2018 dataset carries greater scores.

4.3. State-of-the-art comparison

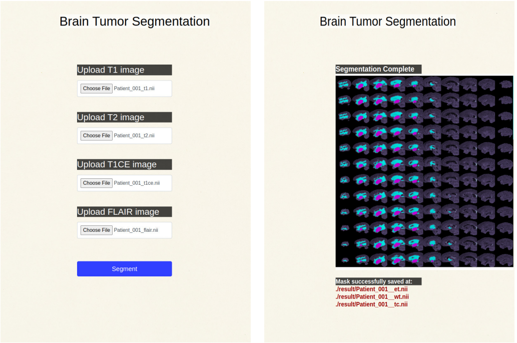
Table 5 represents a comparison between the efficiency of our network and other well-known articles that trained their segmentation networks using the BraTS2020 dataset. The articles highlighted in the table utilize almost the entire dataset and incorporate their version of NN-based algorithms. Most of these articles enhance their accuracy with the help of elaborate pre-processing and multi-level image-capturing techniques. However, our simulation results are mostly higher in comparison to others since GSNet scores the highest among all the articles in terms of WT and TC class segmentation. In terms of ET, it is the second best result, after article [51] with only a 0.38% difference. Some of these articles used multi-stage and multi-stacked UNet-type structures, for instance, articles [17] and [18]. Some of them even pre-processed the 3D images in 2D patches [50]. However, our network uses around half the dataset for training and still produces better scores without extensive pre-processing. Based on the DSC values obtained, it is safe to designate GSNet as a state-of-the-art segmentation network.

Table 5. DSC score comparison with various articles utilizing their versions of UNet with BraTS 2020 dataset

Name	Method	Validation DSC		
		WT	TC	ET
[15]	The authors utilized a 2D convolutional neural network to create a basic encoder-decoder-based UNet-type structure.	0.8674	0.7983	0.7514
[18]	The authors used separate encoders for all 4 types of images (FLAIR, t1, t1ce, and t2) and combined the outputs in a single decoder.	0.7025 (Intact Tumor)	0.8827	0.7386
[49]	The article shows a multiple-stage solution where a 3D dense convolutional block is used in combination with residual inception blocks. The model utilized the entire dataset.	0.8912	0.8474	0.7912
[50]	The authors used a 2D axial path-wise approach for segmentation. Their network contains 2D fully convolutional networks and variants of deep-layer aggregation units stacked together. Their 2D input patches are sized 120 × 120.	0.8858	0.8297	0.7900
[51]	The authors used all 369 cases from the entire dataset. They proposed a new architecture that is made with SANet (scale attention mechanisms) and combined it with a residual squeeze and excitation network utilizing 24 features. Their input contains randomly cropped (128 × 128 × 128) images. Their proposed model is the 3rd place solution in the BraTS 2020 challenge.	0.8828	0.8433	**0.8177**
[17]	The authors proposed a parallel multi-scale fusion module and expectation maximization attention mechanism. Overall it is a 2-stage cascaded 3D UNet-type model.	0.9129	0.8546	0.7875
GSNet	We have utilized multi-input attention-based skip connection combined with our proposed 3D convolution-based segmentation network.	**0.9239**	**0.9103**	0.8139

4.4. GSNet-based web app

To apply our model as a medical imaging tool, we have compressed the entire pipeline in an easy-to-use web-based application (web app) [52] which takes in input as 3D MRI images in the format of '.nii' files. With the input of FLAIR, t1, t1ce, and t2 MRI scans, our web app can produce the corresponding segmentation masks for WT, TC, and ET regions and further save them locally in the '.nii' file format in under 20 seconds. Due to the lightweight size, the web app creates the segmentation masks very quickly. The pre-trained weight file corresponding to this web app created from training with the BraTS 2020 dataset is around 24.7MB. The total size of the web app including the weight file is 36.8MB. The outputs which are saved as '.nii' files are around 8MB each so 24MB in total (3 outputs for 3 separate regions). We have deployed the web app on our local system, which runs through a web browser at "http://127.0.0.1:5000/". An example of using our web app is shown in Fig. 17. Figure 17(left) is displaying the input page where FLAIR, t1, t1ce, and t2 MRI scans are given. The web app creates the outputs and saves them in any local folder. The corresponding indication is shown (the red-marked lines at the bottom) on the next page of the web app (Fig. 17(right)). We have visualized the saved outputs using the software ITK-SNAP. An example of the saved outputs is shown below (Fig. 18) [53]. The full installation process along with the necessary files are provided at the GitHub link [54].



Brain Tumor Segmentation

Upload T1 image

Patient_001_t1.nii

Upload T2 image

Patient_001_t2.nii

Upload T1CE image

Patient_001_t1ce.nii

Upload FLAIR image

Patient_001_flair.nii

Brain Tumor Segmentation

Segmentation Complete

Mask successfully saved at:
./result/Patient_001__et.nii
./result/Patient_001__wt.nii
./result/Patient_001__tc.nii

Fig. 17. Our GSNet-based web app. ((left) opening page, (right) result page)

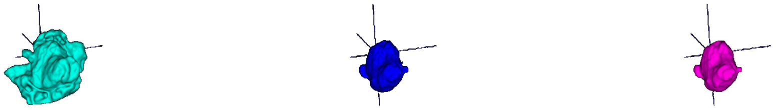


Fig. 18. The regions for each class segmented using the masks saved through the web app. ((left) WT, (middle) TC, and (right) ET)

5. Conclusion

The process of segmentation, if carried out manually can pose a very tiresome struggle. Fortunately, CBIR tools have started to play a significant role in helping medical personnel. In a similar manner, our network GSNet can play a beneficial role in extracting glioma segments. The network reflects a very high efficiency despite dealing with 3D imbalanced multiclass data. With 0.9239, 0.9103, and 0.8139 DSC scores respectively for WT, TC, and ET regions, GSNet carries out a very accurate multi-class segmentation procedure and the entire pipeline directly involves the given volumetric MRI images as it does not require intermediate data manipulation. The current results, especially compared to the state-of-the-art articles, establish GSNet as a reliable segmentation model. The network outperforms existing conventional approaches with ease and produces multi-class glioma segments. The overall lightweight structure of the network helps the model for quick usage and the GSNet-based web app further verifies this by creating the segmentation masks in only 20 seconds.

In future work regarding this research, we would like to tackle some barriers. Despite being very efficient, our model has its limits. The network is constructed primarily for segmentation. However, given that our training data was well-suited, we cannot guarantee its performance on corrupted images. Perhaps, we can explore image restoration strategies to deal with degraded samples. In terms of class imbalance, although the network performed well, we purposefully did not integrate any extensive pre-processing methods. In the future, we can combine class weighting or image augmentation techniques to maintain the balance before training. Furthermore, due to computational limitations, we could not apply higher image dimensions for training. With the addition of more GPUs with higher RAM, we can explore higher resolutions in the future.

In the case of the web app, the primary purpose of creating it was to show GSNet's capability. However, it is not suited for production-level tasks yet, since it lacks the necessary security measures. In the future, we will explore further privacy tools to construct the web app thoroughly. The GSNet model can also be potentially integrated to build a brain tumor patient survival model. This will initially extract the tumors using the GSNet structure and then further classify the ROIs among survived or not survived patients. The necessary resources are also available from the BraTS 2020 dataset and can be utilized further for this survival prediction. Both machine learning and traditional mathematical approaches can be applied to predict the survival rate. Apart from this, the GSNet structure can be tested in segmenting other types of tumors. It can be further fine-tuned for modified tumor-related diagnosis.

Disclosures. The authors declare no conflict of interest.

Data Availability. The data underlying the results presented in this paper are not publicly available at this time but may be obtained from authors upon reasonable request.

References

1. Brain tumor: Statistics. https://www.cancer.net/cancer-types/brain-tumor/statistics.
2. Key statistics for brain and spinal cord tumors. https://www.cancer.org/cancer/brain-spinal-cord-tumors-adults/about/key-statistics.html.
3. S. Minaee, Y. Boykov, F. Porikli, *et al.*, "Image segmentation using deep learning: A survey," IEEE Trans. Pattern Anal. Mach. Intell. **44**(7), 3523–3542 (2021).
4. M. Havaei, A. Davy, D. Warde-Farley, *et al.*, "Brain tumor segmentation with deep neural networks," Med. Image Anal. **35**, 18–31 (2017).
5. A. Işın, C. Direkoğlu, and M. Şah, "Review of MRI-based brain tumor image segmentation using deep learning methods," Procedia Comput. Sci. **102**, 317–324 (2016).
6. S. Tongbram, B. A. Shimray, L. S. Singh, *et al.*, "A novel image segmentation approach using fcm and whale optimization algorithm," Journal of Ambient Intelligence and Humanized Computing pp. 1–15 (2021).
7. M. Abd Elaziz, A. A. Ewees, and D. Oliva, "Hyper-heuristic method for multilevel thresholding image segmentation," Expert Syst. Appl. **146**, 113201 (2020).
8. V. Sivakumar and N. Janakiraman, "A novel method for segmenting brain tumor using modified watershed algorithm in MRI image with FPGA," BioSystems **198**, 104226 (2020).
9. J. Gao, B. Wang, Z. Wang, *et al.*, "A wavelet transform-based image segmentation method," Optik **208**, 164123 (2020).
10. 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*, (Springer, 2015), pp. 234–241.
11. M. Lyksborg, O. Puonti, M. Agn, *et al.*, "An ensemble of 2D convolutional neural networks for tumor segmentation," in *Image Analysis: 19th Scandinavian Conference, SCIA 2015, Copenhagen, Denmark, June 15-17, 2015. Proceedings 19*, (Springer, 2015), pp. 201–211.
12. A. Saha, Y.-D. Zhang, and S. C. Satapathy, "Brain tumour segmentation with a muti-pathway ResNet based UNet," J. Grid Computing **19**(4), 43 (2021).
13. K. He, X. Zhang, S. Ren, *et al.*, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (IEEE, 2016), pp. 770–778.
14. H. Zunair and A. B. Hamza, "Sharp U-Net: Depthwise convolutional network for biomedical image segmentation," Comput. Biol. Med. **136**, 104699 (2021).
15. J. Colman, L. Zhang, W. Duan, *et al.*, "DR-Unet104 for multimodal MRI brain tumor segmentation," 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. 410–419.
16. T. Henry, A. Carré, M. Lerousseau, *et al.*, "Brain tumor segmentation with self-ensembled, deeply-supervised 3D U-net neural networks: A BraTS 2020 challenge solution," 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 I 6*, (Springer, 2021), pp. 327–339.
17. H. Jia, W. Cai, H. Huang, *et al.*, "H2NF-Net for brain tumor segmentation using multimodal MR imaging: 2nd place solution to brats challenge 2020 segmentation task," 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. 58–68.
18. W. Zhang, G. Yang, H. Huang, *et al.*, "ME-Net: Multi-encoder net framework for brain tumor segmentation," Int. J. Imaging Syst. Technol. **31**(4), 1834–1848 (2021).
19. 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)*, (IEEE, 2016), pp. 565–571.
20. W. Wang, C. Chen, M. Ding, *et al.*, "TransBTS: Multimodal brain tumor segmentation using transformer," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I 24*, (Springer, 2021), pp. 109–119.
21. What is a glioma? https://www.hopkinsmedicine.org/health/conditions-and-diseases/gliomas.
22. S. Bakas, H. Akbari, A. Sotiras, *et al.*, "Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features," Sci. Data **4**(1), 170117 (2017).
23. Imaging data description. https://www.med.upenn.edu/cbica/brats2020/data.html.

24. M. Bhagat and B. Bakariya, "Implementation of logistic regression on diabetic dataset using train-test-split, k-fold and stratified k-fold approach," Natl. Acad. Sci. Lett. **45**(5), 401–404 (2022).
25. T.-W. Ke, A. S. Brewster, S. X. Yu, *et al.*, "A convolutional neural network-based screening tool for X-ray serial crystallography," J. Synchrotron Radiat. **25**(3), 655–670 (2018).
26. F. Feng, S. Wang, C. Wang, *et al.*, "Learning deep hierarchical spatial–spectral features for hyperspectral image classification based on residual 3D-2D CNN," Sensors **19**(23), 5276 (2019).
27. S. Ji, W. Xu, M. Yang, *et al.*, "3D convolutional neural networks for human action recognition," IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 221–231 (2013).
28. R. Haldar, L. Wu, J. Xiong, *et al.*, "A multi-perspective architecture for semantic code search," arXiv arXiv:2005.06980 (2020).
29. I. Colbert, K. Kreutz-Delgado, and S. Das, "An energy-efficient edge computing paradigm for convolution-based image upsampling," EEE Access **9**, 147967 (2021).
30. D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," arXiv arXiv:1607.08022 (2016).
31. M. Drozdzal, E. Vorontsov, G. Chartrand, *et al.*, "The importance of skip connections in biomedical image segmentation," in *International Workshop on Deep Learning in Medical Image Analysis, International Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*, (Springer, 2016), pp. 179–187.
32. C. Li, Y. Tan, W. Chen, *et al.*, "ANU-Net: Attention-based nested U-Net to exploit full resolution features for medical image segmentation," Comput. & Graph. **90**, 11–20 (2020).
33. W. Zhang, J. Li, and Z. Hua, "Attention-based tri-UNet for remote sensing image pan-sharpening," IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing **14**, 3719–3732 (2021).
34. A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," Advances in Neural Information Processing Systems **30**, (2017).
35. H. Ide and T. Kurita, "Improvement of learning for CNN with ReLU activation by sparse regularization," in *2017 International Joint Conference on Neural Networks (IJCNN)*, (IEEE, 2017), pp. 2684–2691.
36. S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," Towards Data Science **6**, 310–316 (2017).
37. T. Eelbode, J. Bertels, M. Berman, *et al.*, "Optimization for medical image segmentation: Theory and practice when evaluating with dice score or Jaccard index," IEEE Trans. Med. Imaging **39**(11), 3679–3690 (2020).
38. S. Jadon, "A survey of loss functions for semantic segmentation," in *2020 IEEE conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, (IEEE, 2020), pp. 1–7.
39. T.-Y. Lin, P. Goyal, R. Girshick, *et al.*, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, (IEEE, 2017), pp. 2980–2988.
40. A. W. Setiawan, "Image segmentation metrics in skin lesion: Accuracy, sensitivity, specificity, dice coefficient, Jaccard index, and Matthews correlation coefficient," in *2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, (IEEE, 2020), pp. 97–102.
41. H. Cao, Y. Wang, J. Chen, *et al.*, "Swin-unet: Unet-like pure transformer for medical image segmentation," in *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, (Springer, 2023), pp. 205–218.
42. F. Liu, X. Ren, Z. Zhang, *et al.*, "Rethinking skip connection with layer normalization in transformers and ResNets," arXiv arXiv:2105.07205 (2021).
43. T. Tong, G. Li, X. Liu, *et al.*, "Image super-resolution using dense skip connections," in *Proceedings of the IEEE International Conference on Computer Vision*, (IEEE, 2017), pp. 4799–4807.
44. GSNet structure GitHub link. https://github.com/006jawad/GSNet_/blob/main/GSNet.py.
45. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv arXiv:1412.6980 (2014).
46. M. Gori and A. Tesi, "On the problem of local minima in backpropagation," IEEE Trans. Pattern Anal. Machine Intell. **14**(1), 76–86 (1992).
47. R. Zhao, B. Qian, X. Zhang, *et al.*, "Rethinking dice loss for medical image segmentation," in *2020 IEEE International Conference on Data Mining (ICDM)*, (IEEE, 2020), pp. 851–860.
48. A. M. Carrington, P. W. Fieguth, H. Qazi, *et al.*, "A new concordant partial AUC and partial c statistic for imbalanced data in the evaluation of machine learning algorithms," BMC Med. Inf. Decis. Making **20**(1), 4–12 (2020).
49. P. Ahmad, S. Qamar, L. Shen, *et al.*, "Context aware 3D UNet for brain tumor segmentation," 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 I 6*, (Springer, 2021), pp. 207–218.
50. C. A. Silva, A. Pinto, S. Pereira, *et al.*, "Multi-stage deep layer aggregation for brain tumor segmentation," 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. 179–188.
51. Y. Yuan, "Automatic brain tumor segmentation with scale attention network," 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 I 6*, (Springer, 2021), pp. 285–294.
52. Our GSNet-based Web App. https://youtu.be/5vl5Yezn6C0.

53. Web App output. https://youtu.be/U09Ur23ldjM.
54. Web App installation. https://github.com/006jawad/GSNet_/tree/main/WebApp.